
asap-modules Documentation

Release 0.0.1

Gayathri Mahalingam, Russel Torres, Daniel Kapner, Sharmi Sesh

Jun 15, 2022

CONTENTS:

1	README	1
1.1	ASAP-modules	1
1.2	Installation	1
1.3	How to run	1
1.4	Other Modules	1
1.5	Support	2
1.6	Acknowledgments	2
2	Installation	3
2.1	Prerequisites	3
2.2	Installing ASAP-modules	3
2.3	Install from source	3
2.4	Docker setup	3
2.5	Running docker	4
2.6	Running modules of ASAP	4
3	Lens distortion correction	5
3.1	Step 1 - Compute lens distortion correction	5
3.2	Step 2 - Apply lens correction	5
4	Generate MIPmaps	7
4.1	Step 1 - Generate MIPmaps	7
4.2	Step 2 - Apply MIPmaps to Render stack	7
5	2D Stitching	9
5.1	Step 1 - Create tilepairs	9
5.2	Step 2 - Generate point matches for the serial section	9
5.3	Step 3 - Solve for transformations	10
5.4	Step 4 - Montage QC	10
6	Global 3D non-linear alignment	11
6.1	Step 1 - Montage scapes generation	11
6.2	Step 2 - Generate tilepairs	11
6.3	Step 3 - Generate point matches	11
6.4	Step 4 - Solve for 3D non-linear transformations	12
7	asap-modules	13
7.1	asap package	13
8	Indices and tables	89

Python Module Index	91
Index	93

1.1 ASAP-modules

ASAP is a set of modules to perform stitching and alignment of EM and Array tomography data. It is suitable for processing large-scale datasets and supports multiple computational environments.

1.2 Installation

Please refer the documentation `Installation:Installation.` on how to install and use ASAP modules

1.3 How to run

The order of processing is as follows;

1. *Lens distortion correction*
2. *Mipmap generation*
3. *Montaging and Montage QC*
4. *Global 3D non-linear alignment*

1.4 Other Modules

A few other modules are included in ASAP to do the following.

1. Materialization - render intermediate/final aligned volume to disk for further processing
2. Fusion - Fuse global 3D non-linear aligned chunks together to make a complete volume
3. Point match filter - A module that performs point match filtering of an existing point match collection
4. Point match optimization - Performs a parameter sweep from a given set of ranges on a random sample of tilepairs to identify the optimal set of parameters
5. Registration - Register individual sections in an already aligned volume (useful in cases of aligning missing/reimaged sections)

1.5 Support

We are planning on occasional updating this tool with no fixed schedule. Community involvement is encouraged through both issues and pull requests. Please make pull requests against the dev branch, as we will test changes there before merging into master.

1.6 Acknowledgments

This project is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior / Interior Business Center (DoI/IBC) contract number D16PC00004. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

INSTALLATION

2.1 Prerequisites

ASAP requires Render web service (<https://github.com/saalfeldlab/render>) to be installed for storing and processing the data. Please refer to [Render](#) for details on its installation.

2.2 Installing ASAP-modules

ASAP can be installed using the following commands.

Clone this repository

```
# git clone this repository
git clone https://github.com/AllenInstitute/asap-modules .
```

2.3 Install from source

CD to the cloned directory and run setup.py

```
python setup.py install
```

2.4 Docker setup

You can also install asap-modules using the provided docker file

```
docker build -t asap-modules:latest --target asap-modules .
```

2.5 Running docker

The built docker image can then be run using the following command

```
docker run --rm asap-modules:latest
```

2.6 Running modules of ASAP

Once ASAP is installed using the above command, you can use all of its functionalities as follows;

```
python -m asap.<submodule>.<function_to_run> --input_json <input_json_file.json> --  
↪output_json <output_json_file.json>
```

For example, the montage qc module can be run using the following command.

```
python -m asap.em_montage_qc.detect_montage_defects --input_json <your_input_json_file_  
↪with_required_parameters> --output_json <output_json_file_with_full_path>
```

and here is an example input json file for the detect_montage_defects module

```
{  
  "render":{  
    "host": <render_host>,  
    "port": <render_port>,  
    "owner": <render_project_owner>,  
    "project": <render_project_name>,  
    "client_scripts": <path_to_render_client_scripts>  
  },  
  "prestitched_stack": <pre_montage_stack>,  
  "poststitched_stack": <montaged_stack>,  
  "match_collection_owner": <owner_of_point_match_collection>,  
  "match_collection": <name_of_point_match_collection>,  
  "out_html_dir": <path_to_directory_to_store_the_qc_plot_html_file>,  
  "plot_sections": <True/False>,  
  "minZ": <z_index_of_the_first_section_to_run_qc_for>,  
  "maxZ": <z_index_of_the_last_section_to_run_qc_for>,  
  "neighbors_distance": <qc_parameter>,  
  "min_cluster_size": <qc_parameter>,  
  "residual_threshold": <qc_parameter>,  
  "pool_size": <pool_size_for_parallel_processing>  
}
```

The list of parameters required for each module can be found out using the `--help` option.

```
# find the list of parameters for the solver module using its help option  
python -m asap.solver.solve --help
```


LENS DISTORTION CORRECTION

The lens distortion correction transforms can be computed using the following modules

3.1 Step 1 - Compute lens distortion correction

Compute lens distortion correction transformation

(Assumes that the images for computation are loaded into a render stack.)

```
python -m asap.mesh_lens_correction.do_mesh_lens_correction --input_json <input_
↪parameter_json_file> --output_json <output_json_file>
```

An example input json file is provided in the do_mesh_lens_correction.py file

3.2 Step 2 - Apply lens correction

Apply lens correction transformations to the input render stack (update the raw tilespecs)

```
python -m asap.lens_correction.apply_lens_correction --input_json <input_parameter_json_
↪file> --output_json <output_json_file>
```

An example input json file is provided in the apply_lens_correction.py file

GENERATE MIPMAPS

MIPmaps are essential for stitching and alignment and is used to generate point matches, used for visualization, etc. MIPmaps can be generated for a dataset loaded into a render stack.

4.1 Step 1 - Generate MIPmaps

```
python -m asap.dataimport.create_mipmaps --input_json <input_parameter_json_file> --  
↪output_json <output_json_file>
```

4.2 Step 2 - Apply MIPmaps to Render stack

```
python -m asap.dataimport.apply_mipmaps_to_render --input_json <input_parameter_json_  
↪file> --output_json <output_json_file>
```

Example input parameter json files are included in the module's script files.

2D STITCHING

2D stitching of serial sections involves the following process

5.1 Step 1 - Create tilepairs

```
python -m asap.pointmatch.create_tilepairs --input_json <input_parameter_json_file> --  
↪ output_json <output_json_file>
```

5.2 Step 2 - Generate point matches for the serial section

ASAP utilizes the SIFT point matching module in Render to compute the point matches. There also exists an opencv version of SIFT computation in ASAP.

5.2.1 Point matching implementation from Render to be run on Spark cluster

This requires Spark to be installed in the setup.

```
python -m asap.pointmatch.generate_point_matches_spark --input_json <input_parameter_  
↪ json_file> --output_json <output_json_file>
```

5.2.2 Point matching implementation from Render to be run on PBS cluster

```
python -m asap.pointmatch.generate_point_matches_qsub --input_json <input_parameter_json_  
↪ file> --output_json <output_json_file>
```

5.2.3 Point matching implementation using opencv

```
python -m asap.pointmatch.generate_point_matches_opencv --input_json <input_parameter_  
↪ json_file> --output_json <output_json_file>
```

The point matches will be saved in a point match collection in the Render web service.

5.3 Step 3 - Solve for transformations

The bigfeta solver can be invoked from asap to solve for transformations using the following command

```
python -m asap.solver.solve --input_json <input_parameter_json_file> --output_json  
↪ <output_json_file>
```

5.4 Step 4 - Montage QC

The solver writes the transformations in the tilespecs associated with the serial section in the render stack. Once this is done, the QC module can be run to gather statistics about the quality of the stitching and also visualization plots of the stitched section.

```
python -m asap.em_montage_qc.detect_montage_defects --input_json <input_parameter_json_  
↪ file> --output_json <output_json_file>
```

The QC plots will be saved in the output directory specified in the input_parameter_json_file and the sections with issues will be found in the output_json_file.

GLOBAL 3D NON-LINEAR ALIGNMENT

Global 3D non-linear alignment can be performed on a stack in chunks as well as the entire dataset (if all the serial sections are montaged and available). The following steps illustrate the global 3D non-linear alignment process.

6.1 Step 1 - Montage scapes generation

Montage scapes are downsampled versions of the serial sections and are used in the global 3D alignment process. Montage scapes can be generated as follows.

```
# Generate downsampled versions of montaged serial sections
python -m asap.materialize.render_downsample_sections --input_json <input_parameter_json_
↪file> --output_json <output_json_file>
```

```
# Create a downsampled montage stack
python -m asap.dataimport.make_montage_scapes_stack --input_json <input_parameter_json_
↪file> --output_json <output_json_file>
```

6.2 Step 2 - Generate tilepairs

3D tilepairs for the downsampled stack can be generated using the following command.

```
python -m asap.pointmatch.create_tilepairs --input_json <input_parameter_json_file> --
↪output_json <output_json_file>
```

6.3 Step 3 - Generate point matches

3D point matches can be generated using the generated tile pairs. The following command can be used to generate point matches using a Spark cluster

```
python -m asap.pointmatch.generate_point_matches_using_spark --input_json <input_
↪parameter_json_file> --output_json <output_json_file>
```

6.4 Step 4 - Solve for 3D non-linear transformations

This step in practice is done as a multi-step 3D alignment process, where a series of transformations (rigid, affine, non-linear) are computed and used as initialization for the computation of next higher order transformation.

A mesh based alignment can also be applied as a last step and is available in [Bigfeta](#).

The command to run the solver is shown below.

```
python -m asap.solver.solve --input_json <input_parameter_json_file> --output_json  
↪ <output_json_file>
```

NOTE: Each of the modules' script include an example input json file for reference and the list of input and output parameters can also be listed using the `-help` option in each of the above commands.

ASAP-MODULES

7.1 asap package

7.1.1 Subpackages

`asap.dataimport` package

Submodules

`asap.dataimport.apply_mipmaps_to_render` module

`class` `asap.dataimport.apply_mipmaps_to_render.AddMipMapsToStack`(*schema_type=None*, *args, **kwargs)

Bases: *StackTransitionModule*

Note: This class takes a *ArgSchema* as an input to parse inputs , with a default schema of type *AddMipMapsToStackParameters*

default_output_schema

alias of *AddMipMapsToStackOutput*

default_schema

alias of *AddMipMapsToStackParameters*

run()

`asap.dataimport.apply_mipmaps_to_render.addMipMapsToRender`(*render, input_stack, mipmap_prefix, imgformat, levels, z*)

asap.dataimport.create_mipmaps module

exception asap.dataimport.create_mipmaps.**CreateMipMapException**

Bases: [*RenderModuleException*](#)

Exception raised when there is a problem creating a mipmap

asap.dataimport.create_mipmaps.**create_mipmaps**(*inputImage*, *outputDirectory*='.', *args, **kwargs)

legacy create_mipmaps function

asap.dataimport.create_mipmaps.**create_mipmaps_legacy**(*inputImage*, *outputDirectory*='.', *args, **kwargs)

legacy create_mipmaps function

asap.dataimport.create_mipmaps.**create_mipmaps_uri**(*inputImage*, *outputDirectory*=None, *method*='block_reduce', *mipmaplevels*=[1, 2, 3], *outputformat*='tif', *convertTo8bit*=True, *force_redo*=True, **kwargs)

function to create downsampled images from an input image

Parameters

- **inputImage** (*str*) – path to input image
- **outputDirectory** (*str*) – path to save output images (default to current directory)
- **mipmaplevels** (*list or tuple*) – list or tuple of integers (default to (1,2,3))
- **outputformat** (*str*) – string representation of extension of image format (default tif)
- **convertTo8bit** (*boolean*) – whether to convert the image to 8 bit, dividing each value by 255
- **force_redo** (*boolean*) – whether to recreate mip map images if they already exist
- **method** (*str*) – string corresponding to downsampling method
- **block_func** (*str*) – string corresponding to function used by block_reduce
- **ds_filter** (*str*) – string corresponding to PIL downsample mode

Returns

list of output images created in order of levels

Return type

list

Raises

MipMapException – if an image cannot be created for some reason

asap.dataimport.create_mipmaps.**mipmap_PIL**(*im*, *levels_file_map*, *ds_filter*='NEAREST', *force_redo*=True, **kwargs)

asap.dataimport.create_mipmaps.**mipmap_block_reduce**(*im*, *levels_file_map*, *block_func*='mean', *force_redo*=True, **kwargs)

asap.dataimport.create_mipmaps.**writeImage**(*img*, *outpath*, *force_redo*)

asap.dataimport.generate_EM_tilespecs_from_metafile module

create tilespecs from TEMCA metadata file

```
class asap.dataimport.generate_EM_tilespecs_from_metafile.GenerateEMTileSpecsModule(schema_type=None,
                                                                                      *args,
                                                                                      **kwargs)
```

Bases: *StackOutputModule*

Note: This class takes a *ArgSchema* as an input to parse inputs , with a default schema of type *GenerateEMTileSpecsParameters*

default_output_schema

alias of *GenerateEMTileSpecsOutput*

default_schema

alias of *GenerateEMTileSpecsParameters*

static image_coords_from_stage(stage_coords, resX, resY, rotation)

run()

static sectionId_from_z(z)

tileId_from_basename(fname)

ts_from_imgdata(imgdata, imgprefix, x, y, minint=0, maxint=255, maskUrl=None, width=3840, height=3840, z=None, sectionId=None, scopeId=None, cameraId=None, pixelsize=None)

asap.dataimport.generate_mipmaps module

```
class asap.dataimport.generate_mipmaps.GenerateMipMaps(schema_type=None, *args, **kwargs)
```

Bases: *StackInputModule*

Note: This class takes a *ArgSchema* as an input to parse inputs , with a default schema of type *GenerateMipMapsParameters*

default_output_schema

alias of *GenerateMipMapsOutput*

default_schema

alias of *GenerateMipMapsParameters*

run()

```
asap.dataimport.generate_mipmaps.create_mipmap_from_tuple(mipmap_tuple, levels=[1, 2, 3],
                                                           imgformat='tif', convertTo8bit=True,
                                                           force_redo=True, **kwargs)
```

```
asap.dataimport.generate_mipmaps.create_mipmap_from_tuple_uri(mipmap_tuple, levels=[1, 2, 3],
                                                             imgformat='tif',
                                                             convertTo8bit=True,
                                                             force_redo=True, **kwargs)
```

```
asap.dataimport.generate_mipmaps.get_filepath_from_tilespec(ts)
```

```
asap.dataimport.generate_mipmaps.make_tilespecs_and_cmds(render, inputStack, output_prefix, zvalues,
                                                         levels, imgformat, convert_to_8bit,
                                                         force_redo, pool_size, method)
```

asap.dataimport.make_montage_scapes_stack module

```
class asap.dataimport.make_montage_scapes_stack.MakeMontageScapeSectionStack(schema_type=None,
                                                                              *args,
                                                                              **kwargs)
```

Bases: [StackOutputModule](#)

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type [MakeMontageScapeSectionStackParameters](#)

default_output_schema

alias of [MakeMontageScapeSectionStackOutput](#)

default_schema

alias of [MakeMontageScapeSectionStackParameters](#)

run()

```
asap.dataimport.make_montage_scapes_stack.create_montage_scape_tile_specs(render, input_stack,
                                                                           image_directory,
                                                                           scale, project,
                                                                           tagstr, imgformat,
                                                                           Z,
                                                                           apply_scale=False,
                                                                           uuid_prefix=True,
                                                                           uuid_prefix_length=10,
                                                                           **kwargs)
```

asap.dataimport.schemas module

```
class asap.dataimport.schemas.AddMipMapsToStackOutput(extra=None, only=None, exclude=(),
                                                         prefix="", strict=None, many=False,
                                                         context=None, load_only=(), dump_only=(),
                                                         partial=False)
```

Bases: [DefaultSchema](#)

Table 1: AddMipMapsToStackOutput

key	description	default	field_type	json_type
output_stack	no description	(RE-REQUIRED)	String	str
missing_ts_zs	Z values for which apply mipmaps failed	[]	List	int

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.dataimport.schemas.AddMipMapsToStackParameters(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: [StackTransitionParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 2: AddMipMapsToStackParameters

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	RenderClientParameters
input_stack	no description	(RE-REQUIRED)	String	str
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
overwrite_zlayer	no description	False	Boolean	bool
output_stackVersion	no description	NA	RenderStackVersion	RenderStackVersion
mipmap_dir	directory to which the mipmaps will be stored	NA	InputDir	str
mipmap_prefix	uri prefix from which mipmap locations are built.	(RE-REQUIRED)	String	str
levels	number of levels of mipmaps, default is 6	6	Integer	int
imgformat	mipmap image format, default is tiff	tiff	String	str

mipmap_directory_to_prefix(data)

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.dataimport.schemas.GenerateEMTileSpecsOutput(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 3: GenerateEMTileSpecsOutput

key	description	default	field_type	json_type
stack	stack to which generated tiles were added	(RE-REQUIRED)	String	str

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.dataimport.schemas.GenerateEMTileSpecsParameters(extra=None, only=None, exclude=(),  
                                                             prefix="", strict=None, many=False,  
                                                             context=None, load_only=(),  
                                                             dump_only=(), partial=False)
```

Bases: [OutputStackParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 4: GenerateEMFileSpecsParameters

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
over-write_zlayer	no description	False	Boolean	bool
output_stackVersion	no description	NA	RenderStackValidation	
metafile	metadata file containing TEMCA acquisition data	NA	InputFile	str
metafile_uri	uri of metadata containing TEMCA acquisition data	(RE-REQUIRED)	String	str
maskUrl	absolute path to image mask to apply	None	InputFile	str
maskUrl_uri	uri of image mask to apply	None	String	str
image_directory	directory used in determining absolute paths to images. Defaults to parent directory containing metafile if omitted.	NA	InputDir	str
image_prefix	prefix used in determining full uris of images in metadata. Defaults to using the / delimited prefix to the metadata_uri if omitted	NA	String	str
maximum_intensity	intensity value to interpret as white	255	Integer	int
minimum_intensity	intensity value to interpret as black	0	Integer	int
sectionId	sectionId to apply to tiles during ingest. If unspecified will default to a string representation of the float value of z_index.	NA	String	str

image_directory_to_prefix(data)

maskUrl_to_uri(data)

metafile_to_uri(data)

opts = <marshmallow.schema.SchemaOpts object>

```
class asap.dataimport.schemas.GenerateMipMapsOutput(extra=None, only=None, exclude=(), prefix="",
strict=None, many=False, context=None,
load_only=(), dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 5: GenerateMipMapsOutput

key	description	default	field_type	json_type
levels	no description	(RE-REQUIRED)	Integer	int
output_prefix	no description	(RE-REQUIRED)	String	str

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.dataimport.schemas.GenerateMipMapsParameters(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: [InputStackParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 6: GenerateMipMapsParameters

key	description	default	field_type	json_type
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
input_stack	no description	(RE-REQUIRED)	String	str
output_dir	directory to which the mipmaps will be stored	NA	String	str
output_prefix	uri prefix for generated mipmaps	(RE-REQUIRED)	String	str
method	method to downsample mipmapLevels, 'PIL' for PIL Image (currently NEAREST) filtered resize, can be 'block_reduce' for skimage based area downsampling	block_reduce	String	str
convert_to_8bit	convert the data from 16 to 8 bit (default True)	True	Boolean	bool
pool_size	number of cores to be used	20	Integer	int
imgformat	image format for mipmaps (default tiff)	tiff	String	str
levels	number of levels of mipmaps, default is 6	6	Integer	int
force_redo	force re-generation of existing mipmaps	True	Boolean	bool
PIL_filter	filter to be used in PIL resize	NEAREST	String	str
block_func	function to represent blocks in area downsampling with block_reduce	mean	String	str

directory_to_prefix(data)

`opts = <marshmallow.schema.SchemaOpts object>`

`classmethod validationOptions(options)`

```
class asap.dataimport.schemas.MakeMontageScapeSectionStackOutput(extra=None, only=None,
                                                                    exclude=(), prefix="",
                                                                    strict=None, many=False,
                                                                    context=None, load_only=(),
                                                                    dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 7: MakeMontageScapeSectionStackOutput

key	description	default	field_type	json_type
output_stack	Name of the downsampled sections stack	(RE-REQUIRED)	String	str

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.dataimport.schemas.MakeMontageScapeSectionStackParameters(extra=None, only=None,
                                                                        exclude=(), prefix="",
                                                                        strict=None, many=False,
                                                                        context=None,
                                                                        load_only=(),
                                                                        dump_only=(),
                                                                        partial=False)
```

Bases: [OutputStackParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 8: MakeMontageScapeSectionStackParameters

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
over-write_zlayer	no description	False	Boolean	bool
output_stackVersion	no description	NA	RenderStackVersion	
montage_stack	stack to make a downsample version of	(RE-REQUIRED)	String	str
image_directory	directory that stores the montage scapes	(RE-REQUIRED)	String	str
set_new_z	set to assign new z values starting from 0 (default - False)	False	Boolean	bool
new_z_start	new starting z index	0	Integer	int
remap_section_ids	change section ids to new z values. default = False	False	Boolean	bool
imgformat	image format of the montage scapes (default - tif)	tif	String	str
scale	scale of montage scapes	(RE-REQUIRED)	Float	float
apply_scale	Do you want to scale the downsample to the size of section? Default = False	False	Boolean	bool
doFilter	whether to apply default filtering when generating missing downsamples	True	Boolean	bool
level	integer mipMapLevel used to generate missing downsamples	1	Integer	int
fillWithNoise	Whether to fill the background pixels with noise when generating missing downsamples	False	Boolean	bool
memGB_materialize	max heap size in GB for materialization	12G	String	str
pool_size_materialize	number of processes to generate missing downsamples	1	Integer	int
filterList-Name	Apply specified filter list to all renderings	NA	String	str
uuid_prefix	Prepend uuid to generated tileIds to prevent collisions	True	Boolean	bool
uuid_length	length of uuid4 string used in uuid prefix	10	Integer	int

```
opts = <marshmallow.schema.SchemaOpts object>
```

`validate_data(data)`

Module contents

`asap.em_montage_qc` package

Submodules

`asap.em_montage_qc.detect_montage_defects` module

`class` `asap.em_montage_qc.detect_montage_defects.DetectMontageDefectsModule`(*schema_type=None*,
**args, **kwargs*)

Bases: *RenderModule*

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type *DetectMontageDefectsParameters*

default_output_schema

alias of *DetectMontageDefectsParametersOutput*

default_schema

alias of *DetectMontageDefectsParameters*

run()

`asap.em_montage_qc.detect_montage_defects.check_status_of_stack`(*render, stack, zvalues*)

`asap.em_montage_qc.detect_montage_defects.detect_disconnected_tiles`(*render, prestitched_stack,*
poststitched_stack, z,
pre_tilespecs=None,
post_tilespecs=None)

`asap.em_montage_qc.detect_montage_defects.detect_distortion`(*render, poststitched_stack, zvalue,*
threshold_cutoff=[0.005, 0.005],
pool_size=20)

`asap.em_montage_qc.detect_montage_defects.detect_seams`(*render, stack, match_collection,*
match_owner, z, residual_threshold=8,
distance=60, min_cluster_size=15,
tspecs=None)

`asap.em_montage_qc.detect_montage_defects.detect_stitching_gaps`(*render, prestitched_stack,*
poststitched_stack, z,
pre_tilespecs=None,
tilespecs=None)

```
asap.em_montage_qc.detect_montage_defects.detect_stitching_mistakes(render, prestitched_stack,
                                                                    poststitched_stack,
                                                                    match_collection,
                                                                    match_collection_owner,
                                                                    threshold_cutoff,
                                                                    residual_threshold,
                                                                    neighbor_distance,
                                                                    min_cluster_size, zvalues,
                                                                    pool_size=20)
```

```
asap.em_montage_qc.detect_montage_defects.get_pre_post_tspecs(render, prestitched_stack,
                                                                poststitched_stack, z)
```

```
asap.em_montage_qc.detect_montage_defects.run_analysis(render, prestitched_stack, poststitched_stack,
                                                         match_collection, match_collection_owner,
                                                         residual_threshold, neighbor_distance,
                                                         min_cluster_size, threshold_cutoff, z)
```

asap.em_montage_qc.plots module

```
asap.em_montage_qc.plots.plot_defects(render, stack, out_html_dir, args)
```

```
asap.em_montage_qc.plots.plot_residual(xs, ys, residual)
```

```
asap.em_montage_qc.plots.plot_section_maps(render, stack, post_tspecs, matches, disconnected_tiles,
                                              gap_tiles, seam_centroids, stats, zvalues,
                                              out_html_dir=None, pool_size=5)
```

```
asap.em_montage_qc.plots.point_match_plot(tilespecsA, matches, tilespecsB=None)
```

asap.em_montage_qc.schemas module

```
class asap.em_montage_qc.schemas.DetectDistortionParameters(extra=None, only=None, exclude=(),
                                                             prefix="", strict=None, many=False,
                                                             context=None, load_only=(),
                                                             dump_only=(), partial=False)
```

Bases: [RenderParameters](#), [ZValueParameters](#), [ProcessPoolParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 9: DetectDistortionParameters

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
input_stacks	List of input stacks	None	List	str
threshold_cutoff	Threshold for MAD cutoff in x and y	[0.005, 0.005]	List	float

input_stacks

```
match_collections = argschema.fields.List(
    argschema.fields.Str, required=True, description='List of montage point match collections')
```

```
match_collection_owners = argschema.fields.List(
    argschema.fields.Str, required=False, default=None, missing=None, description='List of the match
collection owners')
```

```
opts = <marshmallow.schema.SchemaOpts object>
```

threshold_cutoff

```
num_threads = argschema.fields.Int(
    required=False, default=20, description='Number of parallel threads')
```

```
@post_load def add_match_collection_owner(self, data):
```

```
    if data['match_collection_owners'] is None:
        data['match_collection_owners'] = [data['render']['owner']] *
        len(data['match_collections'])
```

```
class asap.em_montage_qc.schemas.DetectDistortionParametersOutput(extra=None, only=None,
    exclude=(), prefix="",
    strict=None, many=False,
    context=None, load_only=(),
    dump_only=(),
    partial=False)
```

Bases: [DefaultSchema](#)

Table 10: DetectDistortionParametersOutput

key	description	default	field_type	json_type
dis-torted_sections	List of sections that are distorted	(RE-REQUIRED)	List	int

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
class asap.em_montage_qc.schemas.DetectMontageDefectsParameters(extra=None, only=None,
                                                                    exclude=(), prefix="",
                                                                    strict=None, many=False,
                                                                    context=None, load_only=(),
                                                                    dump_only=(), partial=False)
```

Bases: [RenderParameters](#), [ZValueParameters](#), [ProcessPoolParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 11: DetectMontageDefectsParameters

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
prestitched_stack	Pre stitched stack (raw stack)	(RE-REQUIRED)	String	str
post-stitched_stack	Stitched montage stack	(RE-REQUIRED)	String	str
match_collection	Name of the montage point match collection	NA	String	str
match_collection_owner	Number of the match collection owner	None	String	str
residual_threshold	threshold value to filter residuals for detecting seams (default = 4)	4	Integer	int
neighbors_distance	distance in pixels to look for neighboring points in seam detection (default = 60)	80	Integer	int
min_cluster_size	minimum number of point matches required in each cluster for taking it into account for seam detection (default = 7)	12	Integer	int
threshold_cutoff	Threshold for MAD cutoff in x and y	[0.005, 0.005]	List	float
plot_sections	Do you want to plot the sections with defects (holes or gaps)? Will plot Bokeh plots in a html file	True	Boolean	bool
out_html_dir	Folder to save the Bokeh plot defaults to /tmp directory	None	InputDir	str

```
add_match_collection_owner(data)
```

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
class asap.em_montage_qc.schemas.DetectMontageDefectsParametersOutput (extra=None, only=None,
                                                                    exclude=(), prefix="",
                                                                    strict=None,
                                                                    many=False,
                                                                    context=None,
                                                                    load_only=(),
                                                                    dump_only=(),
                                                                    partial=False)
```

Bases: [DefaultSchema](#)

Table 12: DetectMontageDefectsParametersOutput

key	description	default	field_type	json_type
output_html	Output html file with Bokeh plots showing holes and stitching gaps	None	List	str
qc_passed_sections	List of sections that passed QC	(REQUIRED)	List	int
hole_sections	List of z values which failed QC and has holes	(REQUIRED)	List	int
gap_sections	List of z values which have stitching gaps	(REQUIRED)	List	int
seam_sections	List of z values which have seams detected	(REQUIRED)	List	int
seam_centroids	An array of (x,y) positions of seams for each section with seams	(REQUIRED)	NumpyArray	?
distorted_sections	List of sections that are distorted	(REQUIRED)	List	int

opts = [<marshmallow.schema.SchemaOpts object>](#)

```
class asap.em_montage_qc.schemas.RoughQCOutputSchema (extra=None, only=None, exclude=(), prefix="",
                                                                    strict=None, many=False, context=None,
                                                                    load_only=(), dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 13: RoughQCOutputSchema

key	description	default	field_type	json_type
iou_plot	Pdf/html file showing IOU plots	(REQUIRED)	OutputFile	str
distortion_plot	Pdf/html file with distortion plots	(REQUIRED)	OutputFile	str

opts = [<marshmallow.schema.SchemaOpts object>](#)

```
class asap.em_montage_qc.schemas.RoughQCSchema (extra=None, only=None, exclude=(), prefix="",
                                                                    strict=None, many=False, context=None,
                                                                    load_only=(), dump_only=(), partial=False)
```

Bases: [RenderParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 14: RoughQCSchema

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
input_downsampled_stack	Pre rough aligned downsampled stack	(RE-REQUIRED)	String	str
output_downsampled_stack	Rough aligned stack name	(RE-REQUIRED)	String	str
minZ	min z	(RE-REQUIRED)	Integer	int
maxZ	max z	(RE-REQUIRED)	Integer	int
pool_size	Pool size	10	Integer	int
output_dir	temp filename to save fig	None	OutputDir	str
out_file_format	Do you want the output to be bokeh plots in html (option = 'html') or pdf files for plots (option = 'pdf', default)	pdf	String	str

```
opts = <marshmallow.schema.SchemaOpts object>
```

Module contents

asap.intensity_correction package

Submodules

asap.intensity_correction.apply_multiplicative_correction module

```
class asap.intensity_correction.apply_multiplicative_correction.MultIntensityCorr(schema_type=None,
                                                                                  *args,
                                                                                  **kwargs)
```

Bases: [StackTransitionModule](#)

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type [MultIntensityCorrParams](#)

default_schema

alias of [MultIntensityCorrParams](#)

run()

```
asap.intensity_correction.apply_multiplicative_correction.getImage(ts, channel=None)
```

Simple function to get the level 0 image of this tilespec as a numpy array

Parameters

- **ts** ([renderapi.tilespec.TileSpec](#)) – tilespec to get images from (presently assumes this is a tiff image whose URL can be read with tiffle)

- **channel** (*str*) – channel name to get image of, default=None which will default to the non channel image pyramid

Returns

2d numpy array of this image

Return type

numpy.array

```
asap.intensity_correction.apply_multiplicative_correction.intensity_corr(img, ff, clip,  
                                                                    scale_factor,  
                                                                    clip_min, clip_max)
```

utility function to correct an image with a flatfield correction will take *img* and return *img_out* = *img* * max(*ff*) / (*ff* + .0001) converted back to the original type of *img*

Parameters

- **img** (*numpy.array*) – N,M array to correct, could be any type
- **ff** (*numpy.array*) – N,M array of flatfield correction, could be of any type

Returns

N,M numpy array of the same type as *img* but now corrected

Return type

numpy.array

```
asap.intensity_correction.apply_multiplicative_correction.process_tile(C, dirout, stackname,  
                                                                    clip, scale_factor,  
                                                                    clip_min, clip_max,  
                                                                    input_ts,  
                                                                    corr_dict=None)
```

function to correct each tile in the *input_ts* with the matrix *C*, and potentially move the original tiles to a new location.abs

Parameters

- **C** (*numpy.array*) – a 2d numpy array of uint16 or uint8 that represents the correction to apply
- **corr_dict** (*dict* or *None*) – a dictionary with keys of strings of channel names and values of corrections (as with *C*). If *None*, *C* will be applied to each channel, if they exist.
- **dirout** (*str*) – the path to the directory to save all corrected images
- **input_ts** (*renderapi.tilespec.TileSpec*) – the *tilespec* with the tiles to be corrected

```
asap.intensity_correction.apply_multiplicative_correction.write_image(dirout, orig_imageurl,  
                                                                    Res, stackname, z)
```

asap.intensity_correction.calculate_multiplicative_correction module

class asap.intensity_correction.calculate_multiplicative_correction.**MakeMedian**(*schema_type=None*,
**args*,
***kwargs*)

Bases: *RenderModule*

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type *MakeMedianParams*

default_schema

alias of *MakeMedianParams*

run()

asap.intensity_correction.calculate_multiplicative_correction.**getImageFromTilespecs**(*alltilespecs*,
index,
chan-
nel=None)

asap.intensity_correction.calculate_multiplicative_correction.**make_median_image**(*alltilespecs*,
numtiles,
outImage,
pool_size,
chan=None,
gauss_size=10))

asap.intensity_correction.calculate_multiplicative_correction.**randomly_subsample_tilespecs**(*alltilespecs*,
numtiles)

asap.intensity_correction.schemas module

class asap.intensity_correction.schemas.**MakeMedianParams**(*extra=None*, *only=None*, *exclude=()*,
prefix="", *strict=None*, *many=False*,
context=None, *load_only=()*,
dump_only=(), *partial=False*))

Bases: *StackTransitionParameters*

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 15: MakeMedianParams

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
input_stack	no description	(RE-REQUIRED)	String	str
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
over-write_zlayer	no description	False	Boolean	bool
output_stackVersion	no description	NA	RenderStackValidation	
file_prefix	File prefix for median image file that is saved	Median	String	str
output_directory	Output Directory for saving median image	(RE-REQUIRED)	String	str
num_images	Number of images to randomly subsample to generate median	-1	Integer	int

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.intensity_correction.schemas.MultIntensityCorrParams(extra=None, only=None,
                                                                exclude=(), prefix="",
                                                                strict=None, many=False,
                                                                context=None, load_only=(),
                                                                dump_only=(), partial=False)
```

Bases: [StackTransitionParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 16: MultIntensityCorrParams

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
input_stack	no description	(RE-REQUIRED)	String	str
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
over-write_zlayer	no description	False	Boolean	bool
output_stackVersion	no description	NA	RenderStackValidation	
correction_stack	Correction stack (usually median stack for AT data)	(RE-REQUIRED)	String	str
output_directory	Directory for storing Images	(RE-REQUIRED)	OutputDir	str
cycle_number	what cycleNumber to upload for output_stack on render	2	Integer	int
cycle_step_number	what cycleStepNumber to upload for output_stack on render	1	Integer	int
clip	whether to clip values	True	Boolean	bool
scale_factor	scaling value	1.0	Float	float
clip_min	Min Clip value	0	Integer	int
clip_max	Max Clip value	65535	Integer	int

```
opts = <marshmallow.schema.SchemaOpts object>
```

Module contents

asap.lens_correction package

Submodules

asap.lens_correction.apply_lens_correction module

```
class asap.lens_correction.apply_lens_correction.ApplyLensCorrection(schema_type=None,
                                                                    *args, **kwargs)
```

Bases: [StackTransitionModule](#)

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type

*ApplyLensCorrectionParameters***default_output_schema**alias of *ApplyLensCorrectionOutput***default_schema**alias of *ApplyLensCorrectionParameters***run()****asap.lens_correction.lens_correction module****asap.lens_correction.schemas module**

```
class asap.lens_correction.schemas.ApplyLensCorrectionOutput(extra=None, only=None, exclude=(),
                                                             prefix="", strict=None, many=False,
                                                             context=None, load_only=(),
                                                             dump_only=(), partial=False)
```

Bases: *DefaultSchema*

Table 17: ApplyLensCorrectionOutput

key	description	default	field_type	json_type
stack	stack to which transformed tiles were written	(RE-REQUIRED)	<i>String</i>	str
refId	unique identifier string used as reference ID	(RE-REQUIRED)	<i>String</i>	str
missing_ts_zs	Z values for which apply mipmaps failed	[]	<i>List</i>	int

opts = <marshmallow.schema.SchemaOpts object>

```
class asap.lens_correction.schemas.ApplyLensCorrectionParameters(extra=None, only=None,
                                                                    exclude=(), prefix="",
                                                                    strict=None, many=False,
                                                                    context=None, load_only=(),
                                                                    dump_only=(), partial=False)
```

Bases: *StackTransitionParameters*

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 18: ApplyLensCorrectionParameters

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
input_stack	no description	(RE-REQUIRED)	String	str
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
over-write_zlayer	no description	False	Boolean	bool
output_stackVersion	no description	NA	RenderStackVersion	
transform	no description	NA	TransformParameters	
refId	Reference ID to use when uploading transform to render database (Not Implemented)	(RE-REQUIRED)	String	str
labels	labels for the lens correction transform	['lens']	List	str
maskUrl	path to level 0 maskUrl to apply to stack	None	InputFile	str
maskUrl_uri	uri for level 0 mask image to apply	None	String	str

maskUrl_to_uri(data)

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.lens_correction.schemas.TransformParameters(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(), dump_only=(),
    partial=False)
```

Bases: [DefaultSchema](#)

Table 19: TransformParameters

key	description	default	field_type	json_type
type	Transform type as defined in Render Transform Spec. This module currently expects a 'leaf'	(RE-REQUIRED)	String	str
className	mpicbg-compatible className	(RE-REQUIRED)	String	str
dataString	mpicbg-compatible dataString	(RE-REQUIRED)	String	str
metaData	in this schema, otherwise will be stripped	NA	Dict	?

opts = `<marshmallow.schema.SchemaOpts object>`

Module contents

asap.materialize package

Submodules

asap.materialize.materialize_sections module

Materialize Render sections using BetterBox client

exception asap.materialize.materialize_sections.**MaterializeSectionsError**

Bases: *RenderModuleException*

class asap.materialize.materialize_sections.**MaterializeSectionsModule**(*input_data=None, schema_type=None, output_schema_type=None, args=None, logger_name='argschema.argschema_parser'*)

Bases: *SparkModule*

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type *MaterializeSectionsParameters*

default_output_schema

alias of *MaterializeSectionsOutput*

default_schema

alias of *MaterializeSectionsParameters*

classmethod **get_args**(***kwargs*)

override to append to spark call

classmethod **get_materialize_options**(*baseDataUrl=None, owner=None, project=None, stack=None, rootDirectory=None, width=None, height=None, maxLevel=None, fmt=None, maxOverviewWidthAndHeight=None, skipInterpolation=None, binaryMask=None, label=None, createIGrid=None, forceGeneration=None, renderGroup=None, numberOfRenderGroups=None, cleanUpPriorRun=None, filterListName=None, explainPlan=None, maxImageCacheGb=None, zValues=None, **kwargs*)

run()

asap.materialize.render_downsample_sections module

```
class asap.materialize.render_downsample_sections.RenderSectionAtScale(schema_type=None,
                                                                    *args, **kwargs)
```

Bases: [RenderModule](#)

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type [RenderSectionAtScaleParameters](#)

default_output_schema

alias of [RenderSectionAtScaleOutput](#)

default_schema

alias of [RenderSectionAtScaleParameters](#)

```
classmethod downsample_specific_mipmapLevel(zvalues, input_stack=None, level=1, pool_size=1,
                                             image_directory=None, scale=None,
                                             imgformat=None, doFilter=None,
                                             fillWithNoise=None, filterListName=None,
                                             render=None, do_mp=True, bounds=None,
                                             **kwargs)
```

run()

```
class asap.materialize.render_downsample_sections.WithThreadPool(*args, **kwargs)
```

Bases: [ThreadPool](#)

```
asap.materialize.render_downsample_sections.check_stack_for_mipmaps(render, input_stack,
                                                                    zvalues)
```

```
asap.materialize.render_downsample_sections.create_tilespecs_without_mipmaps(render,
                                                                              montage_stack,
                                                                              level, z)
```

return tilespecs missing mipmaplevels above the specified level

asap.materialize.schemas module

```
class asap.materialize.schemas.Bounds(extra=None, only=None, exclude=(), prefix="", strict=None,
                                       many=False, context=None, load_only=(), dump_only=(),
                                       partial=False)
```

Bases: [DefaultSchema](#)

Table 20: Bounds

key	description	default	field_type	json_type
minX	minX of bounds	None	Integer	int
maxX	maxX of bounds	None	Integer	int
minY	minY of bounds	None	Integer	int
maxY	maxY of bounds	None	Integer	int

opts = `<marshmallow.schema.SchemaOpts object>`


```
class asap.materialize.schemas.DeleteMaterializedSectionsOutput(extra=None, only=None,
                                                                exclude=(), prefix="",
                                                                strict=None, many=False,
                                                                context=None, load_only=(),
                                                                dump_only=(), partial=False)
```

Bases: `DefaultSchema`

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.materialize.schemas.DeleteMaterializedSectionsParameters(extra=None, only=None,
                                                                exclude=(), prefix="",
                                                                strict=None, many=False,
                                                                context=None,
                                                                load_only=(),
                                                                dump_only=(),
                                                                partial=False)
```

Bases: `ArgSchema`

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 21: DeleteMaterializedSectionsParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	<code>InputFile</code>	str
output_json	file path to output json file	NA	<code>OutputFile</code>	str
log_level	set the logging level of the module	ERROR	<code>LogLevel</code>	str
minZ	no description	(RE-REQUIRED)	<code>Integer</code>	int
maxZ	no description	(RE-REQUIRED)	<code>Integer</code>	int
basedir	base directory for materialization	(RE-REQUIRED)	<code>InputDir</code>	str
pool_size	size of pool to use to delete files	NA	<code>Integer</code>	int
tilsource	no description	5	<code>Integer</code>	int

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.materialize.schemas.MaterializeSectionsOutput(extra=None, only=None, exclude=(),
                                                                prefix="", strict=None, many=False,
                                                                context=None, load_only=(),
                                                                dump_only=(), partial=False)
```

Bases: `DefaultSchema`

Table 22: MaterializeSectionsOutput

key	description	default	field_type	json_type
zValues	no description	(RE-REQUIRED)	<code>List</code>	int
rootDirectory	no description	(RE-REQUIRED)	<code>InputDir</code>	str
materialized-Directory	no description	(RE-REQUIRED)	<code>InputDir</code>	str

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.materialize.schemas.MaterializeSectionsParameters(extra=None, only=None, exclude=(),
                                                             prefix="", strict=None, many=False,
                                                             context=None, load_only=(),
                                                             dump_only=(), partial=False)
```

Bases: [ArgSchema](#), [MaterializedBoxParameters](#), [ZRangeParameters](#),
[RenderParameters](#)[RenderWebServiceParameters](#), [SparkParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 23: MaterializeSectionsParameters

key	description	default	field_type	json_type
jarfile	spark jar to call java spark command	(RE-REQUIRED)	String	str
className	spark class to call	(RE-REQUIRED)	String	str
driverMemory	spark driver memory (important for local spark)	6g	String	str
memory	Memory required for spark job	NA	String	str
sparkhome	Spark home directory containing bin/spark_submit	(RE-REQUIRED)	InputDir	str
spark_files	list of spark files to add to the spark submit command	NA	List	str
spark_conf	dictionary of key value pairs to add to spark_submit as --conf key=value	NA	Dict	?
masterUrl	spark master url. For local execution local[num_procs,num_retries]	(RE-REQUIRED)	String	str
baseDataUrl	api endpoint url e.g. http://<host>[:port]/render-ws/v1	NA	String	str
owner	owner of target collection	NA	String	str
project	project fo target collection	NA	String	str
render	no description	NA	RenderClientParameters	
minZ	minimum Z integer	NA	Integer	int
maxZ	maximum Z integer	NA	Integer	int
stack	stack fromw which boxes will be materialized	(RE-REQUIRED)	String	str
rootDirectory	directory in which materialization directory structure will be created (structure is <rootDirectory>/<project>/<stack>/<width>x<height>/<mipMapLevel>/<z>/<row>/<col>.<fmt>)	(RE-REQUIRED)	OutputDir	str
width	width of flat rectangular tiles to generate	(RE-REQUIRED)	Integer	int
height	height of flat rectangular tiles to generate	(RE-REQUIRED)	Integer	int
maxLevel	maximum mipMapLevel to generate.	0	Integer	int
fmt	image format to generate mipmaps – PNG if not specified	NA	String	str
max-OverviewWidthAndHeight	maximum pixel size for width or height of overview image. If excluded or 0, no overview generated.	NA	Integer	int
skipInterpolation	whether to skip interpolation (e.g. DMG data)	NA	Boolean	bool

continues on next page

Table 23 – continued from previous page

key	description	default	field_type	json_type
binaryMask	whether to use binary mask (e.g. DMG data)	NA	Boolean	bool
label	whether to generate single color tile labels rather than actual images	NA	Boolean	bool
createIGrid	whether to create an IGrid file	NA	Boolean	bool
forceGeneration	whether to regenerate existing tiles	NA	Boolean	bool
renderGroup	index (1-n) identifying coarse portion of layer to render	NA	Integer	int
numberOfRenderGroups	used in conjunction with renderGroup, total number of groups being used	NA	Integer	int
filterList-Name	Apply specified filter list to all renderings	NA	String	str
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
cleanUpPriorRun	whether to regenerate most recently generated boxes of an identical plan. Useful for rerunning failed jobs.	NA	Boolean	bool
explainPlan	whether to perform a dry run, logging as partition stages are run but skipping materialization	NA	Boolean	bool
maxImage-CacheGb	maximum image cache in GB of tilespec level 0 data to cache per core. Larger values may degrade performance due to JVM garbage collection.	2.0	Float	float
zValues	z indices to materialize	NA	List	int

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.materialize.schemas.RenderSectionAtScaleOutput(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: `DefaultSchema`

Table 24: RenderSectionAtScaleOutput

key	description	default	field_type	json_type
image_directory	Directory in which the downsampled section images are saved	(REQUIRED)	InputDir	str
temp_stack	The temp stack that was used to generate the downsampled sections	(REQUIRED)	String	str

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.materialize.schemas.RenderSectionAtScaleParameters(extra=None, only=None,
    exclude=(), prefix="", strict=None,
    many=False, context=None,
    load_only=(), dump_only=(),
    partial=False)
```

Bases: [RenderParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 25: RenderSectionAtScaleParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	dict
input_stack	Input stack to make the downsample version of	(RE-REQUIRED)	String	str
image_directory	Directory to save the downsampled sections	(RE-REQUIRED)	OutputDir	str
imgformat	Image format (default - png)	png	String	str
doFilter	Apply filtering before rendering	True	Boolean	bool
fillWithNoise	Fill image with noise (default - False)	False	Boolean	bool
scale	scale of the downsampled sections	(RE-REQUIRED)	Float	float
minZ	min Z to create the downsample section from	-1	Integer	int
maxZ	max Z to create the downsample section from	-1	Integer	int
filterList-Name	Apply specified filter list to all renderings	NA	String	str
bounds	no description	None	Bounds	dict
use_stack_bounds	Do you want to use stack bounds while downsampling?. Default=False	False	Boolean	bool
pool_size	number of parallel threads to use	20	Integer	int

`opts = <marshmallow.schema.SchemaOpts object>`

`validate_data(data)`

```
class asap.materialize.schemas.ValidateMaterializationOutput(extra=None, only=None, exclude=(),
                                                             prefix="", strict=None, many=False,
                                                             context=None, load_only=(),
                                                             dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 26: ValidateMaterializationOutput

key	description	default	field_type	json_type
basedir	no description	(RE-REQUIRED)	String	str
failures	no description	(RE-REQUIRED)	List	str

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.materialize.schemas.ValidateMaterializationParameters(extra=None, only=None,
                                                                exclude=(), prefix="",
                                                                strict=None, many=False,
                                                                context=None, load_only=(),
                                                                dump_only=(), partial=False)
```

Bases: [ArgSchema](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 27: ValidateMaterializationParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
minRow	minimum row to attempt to validate tiles. Will attempt to use stack bounds if None	NA	Integer	int
maxRow	no description	NA	Integer	int
minCol	no description	NA	Integer	int
maxCol	no description	NA	Integer	int
minZ	no description	(RE-REQUIRED)	Integer	int
maxZ	no description	(RE-REQUIRED)	Integer	int
ext	no description	png	String	str
basedir	base directory for materialization	(RE-REQUIRED)	InputDir	str
pool_size	size of pool to use to investigate image validity	NA	Integer	int

`opts = <marshmallow.schema.SchemaOpts object>`

Module contents

[asap.module package](#)

[Subpackages](#)

[asap.module.schemas package](#)

[Submodules](#)

[asap.module.schemas.renderclient_schemas module](#)

```
class asap.module.schemas.renderclient_schemas.FeatureExtractionParameters(extra=None,  
                                                                           only=None,  
                                                                           exclude=(),  
                                                                           prefix="",  
                                                                           strict=None,  
                                                                           many=False,  
                                                                           context=None,  
                                                                           load_only=(),  
                                                                           dump_only=(),  
                                                                           partial=False)
```

Bases: [DefaultSchema](#)

Table 28: FeatureExtractionParameters

key	description	default	field_type	json_type
SIFTfdSize	SIFT feature descriptor size – samples per row and column. 8 if excluded or None	NA	Integer	int
SIFTmin-Scale	SIFT minimum scale – minSize * minScale < size < maxSize * maxScale. 0.5 if excluded or None	NA	Float	float
SIFTmaxScale	SIFT maximum scale – minSize * minScale < size < maxSize * maxScale. 0.85 if excluded or None	NA	Float	float
SIFTsteps	SIFT steps per scale octave. 3 if excluded or None	NA	Integer	int

opts = <marshmallow.schema.SchemaOpts object>

```
class asap.module.schemas.renderclient_schemas.FeatureRenderClipParameters(extra=None,  
                                                                           only=None,  
                                                                           exclude=(),  
                                                                           prefix="",  
                                                                           strict=None,  
                                                                           many=False,  
                                                                           context=None,  
                                                                           load_only=(),  
                                                                           dump_only=(),  
                                                                           partial=False)
```

Bases: [DefaultSchema](#)

Table 29: FeatureRenderClipParameters

key	description	default	field_type	json_type
clipWidth	Full scale pixels to include in clipped rendering of LEFT/RIGHT oriented tile pairs. Will not LEFT/RIGHT clip if excluded or None.	NA	Integer	int
clipHeight	Full scale pixels to include in clipped rendering of TOP/BOTTOM oriented tile pairs. Will not TOP/BOTTOM clip if excluded or None.	NA	Integer	int

opts = <marshmallow.schema.SchemaOpts object>

```
class asap.module.schemas.renderclient_schemas.FeatureRenderParameters(extra=None,
                                only=None, exclude=(),
                                prefix="", strict=None,
                                many=False,
                                context=None,
                                load_only=(),
                                dump_only=(),
                                partial=False)
```

Bases: `DefaultSchema`

Table 30: FeatureRenderParameters

key	description	default	field_type	json_type
renderScale	Scale at which image tiles will be rendered. 1.0 (full scale) if excluded or None	NA	<code>Float</code>	float
renderWith-Filter	Render tiles using default filtering (0 and 255 pixel values replaced with integer in U(64, 191), followed by default NormalizeLocalContrast). True if excluded or None	NA	<code>Boolean</code>	bool
renderWith-outMask	Render tiles without mipMapLevel masks. True if excluded or None	NA	<code>Boolean</code>	bool
render-FullScaleWidth	Full scale width for all rendered tiles	NA	<code>Integer</code>	int
render-FullScale-Height	Full scale height for all rendered tiles	NA	<code>Integer</code>	int
fillWithNoise	Fill each canvas image with noise prior to rendering. True if excluded or None	NA	<code>Boolean</code>	bool
renderFil-terListName	Apply specified filter list to all renderings	NA	<code>String</code>	str

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.module.schemas.renderclient_schemas.FeatureStorageParameters(extra=None,
                                only=None,
                                exclude=(), prefix="",
                                strict=None,
                                many=False,
                                context=None,
                                load_only=(),
                                dump_only=(),
                                partial=False)
```

Bases: `DefaultSchema`

Table 31: FeatureStorageParameters

key	description	default	field_type	json_type
rootFeatureDirectory	Root directory for saved feature lists. Features extracted from dynamically rendered canvases if excluded or None.	NA	<code>String</code>	str
requireStoredFeatures	Whether to throw an exception in case features stored in rootFeatureDirectory cannot be found. Missing features are extracted from dynamically rendered canvases if excluded or None	NA	<code>Boolean</code>	bool
maxFeatureCacheGb	Maximum size of feature cache, in GB. 2GB if excluded or None	NA	<code>Integer</code>	int

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.module.schemas.renderclient_schemas.MatchDerivationParameters(extra=None,  
only=None,  
exclude=(), prefix="",  
strict=None,  
many=False,  
context=None,  
load_only=(),  
dump_only=(),  
partial=False)
```

Bases: `DefaultSchema`

Table 32: MatchDerivationParameters

key	description	default	field_type	json_type
matchRod	Ratio of first to second nearest neighbors used as a cutoff in matching features. 0.92 if excluded or None	NA	Float	float
matchModel-Type	Model to match for RANSAC filtering. 'AFFINE' if excluded or None	NA	String	str
matchIterations	RANSAC filter iterations. 1000 if excluded or None	NA	Integer	int
matchMax-Epsilon	no description	NA	Float	float
matchMinInlierRatio	Minimal ratio of inliers to candidates for successful RANSAC filtering. 0.0 if excluded or None	NA	Float	float
matchMinNumInliers	Minimum absolute number of inliers for successful RANSAC filtering. 4 if excluded or None	NA	Integer	int
matchMaxNumInliers	Maximum absolute number of inliers allowed after RANSAC filtering. unlimited if excluded or None	NA	Integer	int
matchMax-Trust	Maximum trust for filtering such that candidates with cost larger than matchMax-Trust * median cost are rejected. 3.0 if excluded or None	NA	Float	float
matchFilter	whether to match one set of matches, or multiple sets. And, whether to keep them separate, or aggregate them. SINGLE_SET if excluded or None.	NA	String	str

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.module.schemas.renderclient_schemas.MatchWebServiceParameters(extra=None,
                                                                           only=None,
                                                                           exclude=(), prefix="",
                                                                           strict=None,
                                                                           many=False,
                                                                           context=None,
                                                                           load_only=(),
                                                                           dump_only=(),
                                                                           partial=False)
```

Bases: [WebServiceParameters](#)

Table 33: MatchWebServiceParameters

key	description	default	field_type	json_type
baseDataUrl	api endpoint url e.g. <a href="http://<host>[:port]/render-ws/v1">http://<host>[:port]/render-ws/v1	NA	String	str
owner	owner of match collection	NA	String	str
collection	match collection name	NA	String	str

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.module.schemas.renderclient_schemas.MaterializedBoxParameters(extra=None,
only=None,
exclude=(), prefix="",
strict=None,
many=False,
context=None,
load_only=(),
dump_only=(),
partial=False)
```

Bases: `DefaultSchema`

Table 34: MaterializedBoxParameters

key	description	default	field_type	json_type
stack	stack fromw which boxes will be materialized	(REQUIRED)	<code>String</code>	str
rootDirectory	directory in which materialization directory structure will be created (structure is <rootDirectory>/<project>/<stack>/<width>x<height>/<mipMapLevel>/<z>/<row>/<col>.<fmt>)	(REQUIRED)	<code>OutputDir</code>	str
width	width of flat rectangular tiles to generate	(REQUIRED)	<code>Integer</code>	int
height	height of flat rectangular tiles to generate	(REQUIRED)	<code>Integer</code>	int
maxLevel	maximum mipMapLevel to generate.	0	<code>Integer</code>	int
fmt	image format to generate mipmaps – PNG if not specified	NA	<code>String</code>	str
max-OverviewWidth and Height	maximum pixel size for width or height of overview image. If excluded or 0, no overview generated.	NA	<code>Integer</code>	int
skipInterpolation	whether to skip interpolation (e.g. DMG data)	NA	<code>Boolean</code>	bool
binaryMask	whether to use binary mask (e.g. DMG data)	NA	<code>Boolean</code>	bool
label	whether to generate single color tile labels rather than actual images	NA	<code>Boolean</code>	bool
createIGrid	whthter to create an IGrid file	NA	<code>Boolean</code>	bool
forceGeneration	whether to regenerate existing tiles	NA	<code>Boolean</code>	bool
renderGroup	index (1-n) identifying coarse portion of layer to render	NA	<code>Integer</code>	int
numberOfRenderGroups	used in conjunction with renderGroup, total number of groups being used	NA	<code>Integer</code>	int
filterList-Name	Apply specified filter list to all renderings	NA	<code>String</code>	str

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.module.schemas.renderclient_schemas.RenderParametersMatchWebServiceParameters(extra=None,
only=None,
ex-
clude=(),
pre-
fix="",
strict=None,
many=False,
con-
text=None,
load_only=(),
dump_only=(),
par-
tial=False)
```

Bases: [MatchWebServiceParameters](#)

Table 35: RenderParametersMatchWebServiceParameters

key	description	default	field_type	json_type
baseDataUrl	api endpoint url e.g. http://<host>[:port]/render-ws/v1	NA	String	str
owner	owner of match collection	NA	String	str
collection	match collection name	NA	String	str
render	no description	NA	RenderClientParameters	

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
validate_options(data)
```

```
class asap.module.schemas.renderclient_schemas.RenderParametersRenderWebServiceParameters(extra=None,
only=None,
ex-
clude=(),
pre-
fix="",
strict=None,
many=False,
con-
text=None,
load_only=(),
dump_only=(),
par-
tial=False)
```

Bases: [RenderWebServiceParameters](#)

Table 36: RenderParametersRenderWebServiceParameters

key	description	default	field_type	json_type
baseDataUrl	api endpoint url e.g. http://<host>[:port]/render-ws/v1	NA	String	str
owner	owner of target collection	NA	String	str
project	project fo target collection	NA	String	str
render	no description	NA	RenderClientParameters	

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
validate_options(data)
```

```
class asap.module.schemas.renderclient_schemas.RenderWebServiceParameters(extra=None,
                                                                              only=None,
                                                                              exclude=(),
                                                                              prefix="",
                                                                              strict=None,
                                                                              many=False,
                                                                              context=None,
                                                                              load_only=(),
                                                                              dump_only=(),
                                                                              partial=False)
```

Bases: [WebServiceParameters](#)

Table 37: RenderWebServiceParameters

key	description	default	field_type	json_type
baseDataUrl	api endpoint url e.g. http://<host>[:port]/render-ws/v1	NA	String	str
owner	owner of target collection	NA	String	str
project	project fo target collection	NA	String	str

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
class asap.module.schemas.renderclient_schemas.WebServiceParameters(extra=None, only=None,
                                                                         exclude=(), prefix="",
                                                                         strict=None, many=False,
                                                                         context=None,
                                                                         load_only=(),
                                                                         dump_only=(),
                                                                         partial=False)
```

Bases: [DefaultSchema](#)

Table 38: WebServiceParameters

key	description	default	field_type	json_type
baseDataUrl	api endpoint url e.g. http://<host>[:port]/render-ws/v1	NA	String	str

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
class asap.module.schemas.renderclient_schemas.ZRangeParameters(extra=None, only=None,
                                                                      exclude=(), prefix="",
                                                                      strict=None, many=False,
                                                                      context=None, load_only=(),
                                                                      dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 39: ZRangeParameters

key	description	default	field_type	json_type
minZ	minimum Z integer	NA	Integer	int
maxZ	maximum Z integer	NA	Integer	int

opts = `<marshmallow.schema.SchemaOpts object>`

asap.module.schemas.schemas module

class `asap.module.schemas.schemas.RenderClientParameters`(*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: `DefaultSchema`

Table 40: RenderClientParameters

key	description	default	field_type	json_type
host	render host	(RE-REQUIRED)	<code>String</code>	str
port	render post integer	(RE-REQUIRED)	<code>Integer</code>	int
owner	render default owner	(RE-REQUIRED)	<code>String</code>	str
project	render default project	(RE-REQUIRED)	<code>String</code>	str
client_scripts	path to render client scripts	(RE-REQUIRED)	<code>String</code>	str
memGB	string describing java heap memory (default 5G)	5G	<code>String</code>	str

opts = `<marshmallow.schema.SchemaOpts object>`

class `asap.module.schemas.schemas.RenderParameters`(*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: `ArgSchema`

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 41: RenderParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	<code>InputFile</code>	str
output_json	file path to output json file	NA	<code>OutputFile</code>	str
log_level	set the logging level of the module	ERROR	<code>LogLevel</code>	str
render	parameters to connect to render server	(RE-REQUIRED)	<code>RenderClientParameters</code>	

opts = `<marshmallow.schema.SchemaOpts object>`

class `asap.module.schemas.schemas.TemplateOutputParameters`(*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: `DefaultSchema`

Table 42: TemplateOutputParameters

key	description	default	field_type	json_type
output_value	an output of the module	(RE-REQUIRED)	String	str

`opts = <marshmallow.schema.SchemaOpts object>`

class `asap.module.schemas.schemas.TemplateParameters`(*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: [RenderParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 43: TemplateParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	dict
example	an example	(RE-REQUIRED)	String	str
default_val	an example with a default	a default value	String	str

`opts = <marshmallow.schema.SchemaOpts object>`

asap.module.schemas.spark_schemas module

class `asap.module.schemas.spark_schemas.SparkOptions`(*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: [DefaultSchema](#)

Table 44: SparkOptions

key	description	default	field_type	json_type
jarfile	spark jar to call java spark command	(RE-REQUIRED)	String	str
className	spark class to call	(RE-REQUIRED)	String	str
driverMemory	spark driver memory (important for local spark)	6g	String	str
memory	Memory required for spark job	NA	String	str
sparkhome	Spark home directory containing bin/spark_submit	(RE-REQUIRED)	InputDir	str
spark_files	list of spark files to add to the spark submit command	NA	List	str
spark_conf	dictionary of key value pairs to add to spark_submit as --conf key=value	NA	Dict	?

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.module.schemas.spark_schemas.SparkParameters(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: [SparkOptions](#)

Table 45: SparkParameters

key	description	default	field_type	json_type
jarfile	spark jar to call java spark command	(RE-REQUIRED)	String	str
className	spark class to call	(RE-REQUIRED)	String	str
driverMemory	spark driver memory (important for local spark)	6g	String	str
memory	Memory required for spark job	NA	String	str
sparkhome	Spark home directory containing bin/spark_submit	(RE-REQUIRED)	InputDir	str
spark_files	list of spark files to add to the spark submit command	NA	List	str
spark_conf	dictionary of key value pairs to add to spark_submit as --conf key=value	NA	Dict	?
masterUrl	spark master url. For local execution local[num_procs,num_retries]	(RE-REQUIRED)	String	str

`opts = <marshmallow.schema.SchemaOpts object>`

asap.module.schemas.stack_schemas module

```
class asap.module.schemas.stack_schemas.InputStackParameters(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: [RenderParameters](#), [ZValueParameters](#), [OverridableParameterSchema](#)

template schema for schemas which take input from a stack

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 46: InputStackParameters

key	description	default	field_type	json_type
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	Dict
input_stack	no description	(RE-REQUIRED)	String	str

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.module.schemas.stack_schemas.OutputStackParameters(extra=None, only=None,
                                                                exclude=(), prefix="", strict=None,
                                                                many=False, context=None,
                                                                load_only=(), dump_only=(),
                                                                partial=False)
```

Bases: [RenderParameters](#), [ZValueParameters](#), [ProcessPoolParameters](#), [OverridableParameterSchema](#)

template schema for writing tilespecs to an output stack

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 47: OutputStackParameters

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	Dict
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
over-write_zlayer	no description	False	Boolean	bool
output_stackVersion	no description	NA	RenderStackValidation	Dict

`opts = <marshmallow.schema.SchemaOpts object>`


```
class asap.module.schemas.stack_schemas.OverridableParameterSchema(extra=None, only=None,
                                                                    exclude=(), prefix="",
                                                                    strict=None, many=False,
                                                                    context=None, load_only=(),
                                                                    dump_only=(),
                                                                    partial=False)
```

Bases: `DefaultSchema`

```
static fix_badkey(data, badkey, goodkey)
```

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
override_input(data)
```

```
class asap.module.schemas.stack_schemas.ProcessPoolParameters(extra=None, only=None,
                                                                exclude=(), prefix="", strict=None,
                                                                many=False, context=None,
                                                                load_only=(), dump_only=(),
                                                                partial=False)
```

Bases: `DefaultSchema`

Table 48: ProcessPoolParameters

key	description	default	field_type	json_type
pool_size	no description	1	<code>Integer</code>	int

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
class asap.module.schemas.stack_schemas.RenderCycle(extra=None, only=None, exclude=(), prefix="",
                                                       strict=None, many=False, context=None,
                                                       load_only=(), dump_only=(), partial=False)
```

Bases: `DefaultSchema`

Table 49: RenderCycle

key	description	default	field_type	json_type
number	no description	NA	<code>Integer</code>	int
stepNumber	no description	NA	<code>Integer</code>	int

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
class asap.module.schemas.stack_schemas.RenderMipMapPathBuilder(extra=None, only=None,
                                                                    exclude=(), prefix="",
                                                                    strict=None, many=False,
                                                                    context=None, load_only=(),
                                                                    dump_only=(), partial=False)
```

Bases: `DefaultSchema`

Table 50: RenderMipMapPathBuilder

key	description	default	field_type	json_type
rootPath	no description	NA	<code>String</code>	str
num-berOfLevels	no description	NA	<code>Integer</code>	int
extension	no description	NA	<code>String</code>	str

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.module.schemas.stack_schemas.RenderStackVersion(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 51: RenderStackVersion

key	description	default	field_type	json_type
createTimes-tamp	no description	NA	String	str
versionNotes	no description	NA	String	str
cycleNumber	no description	NA	Integer	int
cycleStep-Number	no description	NA	Integer	int
stackResolu-tionX	no description	NA	Float	float
stackResolu-tionY	no description	NA	Float	float
stackResolu-tionZ	no description	NA	Float	float
materialized-BoxRootPath	no description	NA	String	str
mipmapPath-BUILDER	no description	NA	RenderMipMapRichBuilder	RichBuilder
cycle	no description	NA	RenderCycle	dict
stackResolu-tionValues	no description	NA	List	int

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.module.schemas.stack_schemas.StackTransitionParameters(extra=None, only=None,
    exclude=(), prefix="",
    strict=None, many=False,
    context=None, load_only=(),
    dump_only=(),
    partial=False)
```

Bases: [OutputStackParameters](#), [InputStackParameters](#)

template schema for schemas which take input from one stack, perform (mostly render-python based) operations on tiles from that stack, and output tiles to another stack.

This schema is designed to be a `schema_type` for an `ArgSchemaParser` object

Table 52: StackTransitionParameters

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
input_stack	no description	(RE-REQUIRED)	String	str
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
over-write_zlayer	no description	False	Boolean	bool
out-put_stackVersion	no description	NA	RenderStackValidation	

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.module.schemas.stack_schemas.ZValueParameters(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: [OverridableParameterSchema](#)

template schema which interprets z values on which to act assumes a hierarchy such that minZ, maxZ are superceded by z which is superceded by zValues.

Table 53: ZValueParameters

key	description	default	field_type	json_type
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int

generate_zValues(*data*)

opts = `<marshmallow.schema.SchemaOpts object>`

Module contents

Submodules

asap.module.render_module module

class asap.module.render_module.**RenderModule**(*schema_type=None, *args, **kwargs*)

Bases: [ArgSchemaParser](#)

Note: This class takes a [ArgSchema](#) as an input to parse inputs , with a default schema of type [RenderParameters](#)

default_schema

alias of [RenderParameters](#)

exception asap.module.render_module.**RenderModuleException**

Bases: [Exception](#)

Base Exception class for render module

class asap.module.render_module.**SparkModule**(*input_data=None, schema_type=None, output_schema_type=None, args=None, logger_name='argschema.argschema_parser'*)

Bases: [ArgSchemaParser](#)

Note: This class takes a [ArgSchema](#) as an input to parse inputs , with a default schema of type [SparkParameters](#)

default_schema

alias of [SparkParameters](#)

classmethod **get_args**(***kwargs*)

override to append to spark call

static **get_cmd_opt**(*v, flag=None*)

static **get_flag_cmd**(*v, flag=None*)

classmethod **get_spark_call**(*masterUrl=None, jarfile=None, className=None, driverMemory=None, memory=None, sparkhome=None, spark_files=None, spark_conf=None, **kwargs*)

classmethod **get_spark_command**(***kwargs*)

run_spark_command(***kwargs*)

static **sanitize_cmd**(*cmd*)

exception asap.module.render_module.**SparkModuleError**

Bases: [RenderModuleException](#)

error thrown by asap spark modules

class asap.module.render_module.**StackInputModule**(*schema_type=None, *args, **kwargs*)

Bases: [RenderModule](#)

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type [InputStackParameters](#)

default_schema

alias of [InputStackParameters](#)

get_inputstack_zs(*input_stack=None, render=None, **kwargs*)

get_overlapping_inputstack_zvalues(*input_stack=None, zValues=None, render=None, **kwargs*)

class asap.module.render_module.**StackOutputModule**(*schema_type=None, *args, **kwargs*)

Bases: [RenderModule](#)

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type [OutputStackParameters](#)

default_schema

alias of [OutputStackParameters](#)

delete_zValues(*zValues=None, output_stack=None, render=None*)

output_tilespecs_to_stack(*tilespecs, output_stack=None, sharedTransforms=None, close_stack=None, overwrite_zlayer=None, render=None, pool_size=None, **kwargs*)

validate_tilespecs(*input_stack, output_stack, z, render=None*)

class asap.module.render_module.**StackTransitionModule**(*schema_type=None, *args, **kwargs*)

Bases: [StackOutputModule](#), [StackInputModule](#)

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type [OutputStackParameters](#)

asap.module.template_module module

class asap.module.template_module.**TemplateModule**(*schema_type=None, *args, **kwargs*)

Bases: [RenderModule](#)

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type [TemplateParameters](#)

default_output_schema

alias of [TemplateOutputParameters](#)

default_schema

alias of [TemplateParameters](#)

run()

Module contents

`asap.montage` package

Submodules

`asap.montage.run_montage_job_for_section` module

`asap.montage.schemas` module

Module contents

`asap.point_match_optimization` package

Submodules

`asap.point_match_optimization.point_match_optimization` module

`asap.point_match_optimization.schemas` module

```
class asap.point_match_optimization.schemas.PointMatchOptimizationParameters(extra=None,  
                                                                              only=None,  
                                                                              exclude=(),  
                                                                              prefix="",  
                                                                              strict=None,  
                                                                              many=False,  
                                                                              context=None,  
                                                                              load_only=(),  
                                                                              dump_only=(),  
                                                                              partial=False)
```

Bases: *RenderParameters*

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 54: PointMatchOptimizationParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	dict
stack	Name of the stack containing the tile pair	(RE-REQUIRED)	String	str
tile_stack	Name of the stack that will hold these two tiles	None	String	str
tileId1	tileId of the first tile in the tile pair	(RE-REQUIRED)	String	str
tileId2	tileId of the second tile in the tile pair	(RE-REQUIRED)	String	str
pool_size	Pool size for parallel processing	10	Integer	int
SIFT_options	no description	(RE-REQUIRED)	SIFT_options	dict
outputDirectory	Parent directory in which subdirectories will be created to store images and point-match results from SIFT	(RE-REQUIRED)	OutputDir	str
url_options	no description	(RE-REQUIRED)	url_options	dict

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.point_match_optimization.schemas.PointMatchOptimizationParametersOutput(extra=None,
only=None,
ex-
clude=(),
pre-
fix="",
strict=None,
many=False,
con-
text=None,
load_only=(),
dump_only=(),
par-
tial=False)
```

Bases: [DefaultSchema](#)

Table 55: PointMatchOptimizationParametersOutput

key	description	default	field_type	json_type
output_html	Output html file that shows all the tilepair plot and results	(RE-REQUIRED)	String	str

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.point_match_optimization.schemas.PtMatchOptimizationParameters(extra=None,
                                                                              only=None,
                                                                              exclude=(),
                                                                              prefix="",
                                                                              strict=None,
                                                                              many=False,
                                                                              context=None,
                                                                              load_only=(),
                                                                              dump_only=(),
                                                                              partial=False)
```

Bases: [RenderParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 56: PtMatchOptimizationParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
stack	Name of the stack containing the tile pair (not the base stack)	(RE-REQUIRED)	String	str
tile_stack	Name of the stack that will hold these two tiles	None	String	str
tilepair_file	Tile pair file	(RE-REQUIRED)	InputFile	str
no_tilepairs_to_test	Number of tilepairs to be tested for optimization - default = 10	10	Integer	int
filter_tilepairs	Do you want filter the tilepair file for pairs that overlap? - default = False	False	Boolean	bool
max_tilepairs_without_matches	How many tilepairs with matches required for selection of optimized parameter set	0	Integer	int
numberOfThreads	Number of threads to run point matching job	5	Integer	int
SIFT_options	no description	(RE-REQUIRED)	SIFT_options	dict
outputDirectory	Parent directory in which subdirectories will be created to store images and point-match results from SIFT	(RE-REQUIRED)	OutputDir	str
url_options	no description	(RE-REQUIRED)	url_options	dict
pool_size	Pool size for parallel processing	10	Integer	int

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
validate_data(data)
```



```

class asap.point_match_optimization.schemas.PtMatchOptimizationParametersOutput(extra=None,
only=None,
exclude=(),
prefix="",
strict=None,
many=False,
con-
text=None,
load_only=(),
dump_only=(),
par-
tial=False)

```

Bases: `DefaultSchema`

Table 57: PtMatchOptimizationParametersOutput

key	description	default	field_type	json_type
output_html	Output html file that shows all the tilepair plot and results	(RE-REQUIRED)	<code>String</code>	str

`opts = <marshmallow.schema.SchemaOpts object>`

```

class asap.point_match_optimization.schemas.SIFT_options(extra=None, only=None, exclude=(),
prefix="", strict=None, many=False,
context=None, load_only=(),
dump_only=(), partial=False)

```

Bases: `DefaultSchema`

Table 58: SIFT_options

key	description	default	field_type	json_type
SIFTfdSize	SIFT feature descriptor size: how many samples per row and column	[8]	List	int
SIFTmaxScale	SIFT maximum scale: minSize * minScale < size < maxSize * maxScale	[0.85]	List	float
SIFTminScale	SIFT minimum scale: minSize * minScale < size < maxSize * maxScale	[0.5]	List	float
SIFTsteps	SIFT steps per scale octave	[3]	List	int
matchIterations	Match filter iterations	[1000]	List	int
matchMaxEpsilon	Minimal allowed transfer error for match filtering	[20.0]	List	float
matchMaxNumInliers	Maximum number of inliers for match filtering	[500]	List	int
matchMaxTrust	Reject match candidates with a cost larger than maxTrust * median cost	[3.0]	List	float
matchMinInlierRatio	Minimal ratio of inliers to candidates for match filtering	[0.0]	List	float
matchMinNumInliers	Minimal absolute number of inliers for match filtering	[10]	List	int
matchModelType	Type of model for match filtering Possible Values: [TRANSLATION, RIGID, SIMILARITY, AFFINE]	['AFFINE']	List	str
matchRod	Ratio of distances for matches	[0.92]	List	float
renderScale	Render canvases at this scale	[0.35]	List	float

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.point_match_optimization.schemas.url_options(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: `DefaultSchema`

Table 59: url_options

key	description	default	field_type	json_type
normalize-ForMatching	normalize for matching	True	Boolean	bool
renderWith-Filter	Render with Filter	True	Boolean	bool
renderWithoutMask	Render without mask	False	Boolean	bool
excludeAll-Transforms	Exclude all transforms	False	Boolean	bool
excludeFirst-TransformAndAllAfter	Exclude first transform and all after	False	Boolean	bool
exclude-TransformsAfterLast	Exclude transforms after last	False	Boolean	bool

```
opts = <marshmallow.schema.SchemaOpts object>
```

Module contents

asap.pointmatch package

Submodules

asap.pointmatch.create_tilepairs module

```
class asap.pointmatch.create_tilepairs.TilePairClientModule(schema_type=None, *args,
                                                            **kwargs)
```

Bases: *RenderModule*

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type *TilePairClientParameters*

```
client_class = 'org.janelia.render.client.TilePairClient'
```

```
client_script_name = 'run_ws_client.sh'
```

```
default_output_schema
```

alias of *TilePairClientOutputParameters*

```
default_schema
```

alias of *TilePairClientParameters*

```
run()
```

asap.pointmatch.generate_point_matches_qsub module

```
class asap.pointmatch.generate_point_matches_qsub.PointMatchClientModuleQsub(schema_type=None,
                                                                                *args,
                                                                                **kwargs)
```

Bases: *RenderModule*

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type *RenderParameters*

```
run()
```

asap.pointmatch.generate_point_matches_spark module

```
class asap.pointmatch.generate_point_matches_spark.PointMatchClientModuleSpark(input_data=None,
                                                                                  schema_type=None,
                                                                                  out-
                                                                                  put_schema_type=None,
                                                                                  args=None,
                                                                                  log-
                                                                                  ger_name='argschema.argschen
```

Bases: [SparkModule](#)

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type [PointMatchClientParametersSpark](#)

default_output_schema

alias of [PointMatchClientOutputSchema](#)

default_schema

alias of [PointMatchClientParametersSpark](#)

classmethod get_args(kwargs)**

override to append to spark call

classmethod get_pointmatch_args(baseDataUrl=None, owner=None, collection=None, pairJson=None, SIFTfdSize=None, SIFTminScale=None, SIFTmaxScale=None, SIFTsteps=None, matchRod=None, matchModelType=None, matchIterations=None, matchMaxEpsilon=None, matchMinInlierRatio=None, matchMinNumInliers=None, matchMaxNumInliers=None, matchMaxTrust=None, maxFeatureCacheGb=None, clipWidth=None, clipHeight=None, renderScale=None, renderWithFilter=None, renderWithoutMask=None, renderFullScaleWidth=None, renderFullScaleHeight=None, fillWithNoise=None, rootFeatureDirectory=None, renderFilterListName=None, requireStoredFeatures=None, matchFilter=None, **kwargs)

run()

asap.pointmatch.generate_point_matches_spark.add_arg(l, argname, args)

asap.pointmatch.generate_point_matches_spark.form_sift_params_list(args)

asap.pointmatch.generate_point_matches_spark.get_host_port_dict_from_url(url)

asap.pointmatch.schemas module

```
class asap.pointmatch.schemas.CollectionId(extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False)
```

Bases: [Schema](#)

Table 60: CollectionId

key	description	default	field_type	json_type
owner	owner of collection	(RE-REQUIRED)	String	str
name	name of collection	(RE-REQUIRED)	String	str

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.pointmatch.schemas.PointMatchClientOutputSchema(extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False)
```

Bases: [Schema](#)

Table 61: PointMatchClientOutputSchema

key	description	default	field_type	json_type
collectionId	collection identifying details	(RE-REQUIRED)	CollectionId	dict
pairCount	number of tile pairs in collection	(RE-REQUIRED)	Integer	int

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.pointmatch.schemas.PointMatchClientParametersQsub(extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False)
```

Bases: [RenderParameters](#), [SIFTPointMatchParameters](#), [SparkOptions](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 62: PointMatchClientParametersQsub

key	description	default	field_type	json_type
jarfile	spark jar to call java spark command	(RE-REQUIRED)	String	str
className	spark class to call	(RE-REQUIRED)	String	str
driverMemory	spark driver memory (important for local spark)	6g	String	str
memory	Memory required for spark job	NA	String	str
sparkhome	Path to the spark home directory	/allen/aibs/pipeline/image_processing/volume_assembly/utls/spark	String	str
spark_files	list of spark files to add to the spark submit command	NA	List	str

continues on next page

Table 62 – continued from previous page

key	description	default	field_type	json_type
spark_conf	dictionary of key value pairs to add to spark_submit as --conf key=value	NA	Dict	?
baseDataUrl	api endpoint url e.g. http://<host>[:port]/render-ws/v1	NA	String	str
owner	owner of match collection	NA	String	str
collection	match collection name	NA	String	str
render	parameters to connect to render server	(REQUIRED)	RenderClientParameters	dict
matchRod	Ratio of first to second nearest neighbors used as a cutoff in matching features. 0.92 if excluded or None	NA	Float	float
matchModel-Type	Model to match for RANSAC filtering. 'AFFINE' if excluded or None	NA	String	str
matchIterations	RANSAC filter iterations. 1000 if excluded or None	NA	Integer	int
matchMax-Epsilon	no description	NA	Float	float
matchMinInlierRatio	Minimal ratio of inliers to candidates for successful RANSAC filtering. 0.0 if excluded or None	NA	Float	float
matchMinNumInliers	Minimum absolute number of inliers for successful RANSAC filtering. 4 if excluded or None	NA	Integer	int
matchMaxNumInliers	Maximum absolute number of inliers allowed after RANSAC filtering. unlimited if excluded or None	NA	Integer	int
matchMaxTrust	Maximum trust for filtering such that candidates with cost larger than matchMaxTrust * median cost are rejected. 3.0 if excluded or None	NA	Float	float
matchFilter	whether to match one set of matches, or multiple sets. And, whether to keep them separate, or aggregate them. SINGLE_SET if excluded or None.	NA	String	str
rootFeatureDirectory	Root directory for saved feature lists. Features extracted from dynamically rendered canvases if excluded or None.	NA	String	str
require-StoredFeatures	Whether to throw an exception in case features stored in rootFeatureDirectory cannot be found. Missing features are extracted from dynamically rendered canvases if excluded or None	NA	Boolean	bool
maxFeature-CacheGb	Maximum size of feature cache, in GB. 2GB if excluded or None	NA	Integer	int
clipWidth	Full scale pixels to include in clipped rendering of LEFT/RIGHT oriented tile pairs. Will not LEFT/RIGHT clip if excluded or None.	NA	Integer	int

continues on next page

Table 62 – continued from previous page

key	description	default	field_type	json_type
clipHeight	Full scale pixels to include in clipped rendering of TOP/BOTTOM oriented tile pairs. Will not TOP/BOTTOM clip if excluded or None.	NA	Integer	int
renderScale	Scale at which image tiles will be rendered. 1.0 (full scale) if excluded or None	NA	Float	float
renderWith-Filter	Render tiles using default filtering (0 and 255 pixel values replaced with integer in U(64, 191), followed by default NormalizeLocalContrast). True if excluded or None	NA	Boolean	bool
renderWith-outMask	Render tiles without mipMapLevel masks. True if excluded or None	NA	Boolean	bool
render-FullScaleWidth	Full scale width for all rendered tiles	NA	Integer	int
render-FullScale-Height	Full scale height for all rendered tiles	NA	Integer	int
fillWithNoise	Fill each canvas image with noise prior to rendering. True if excluded or None	NA	Boolean	bool
renderFilterListName	Apply specified filter list to all renderings	NA	String	str
SIFTfdSize	SIFT feature descriptor size – samples per row and column. 8 if excluded or None	NA	Integer	int
SIFTmin-Scale	SIFT minimum scale – minSize * minScale < size < maxSize * maxScale. 0.5 if excluded or None	NA	Float	float
SIFTmaxScale	SIFT maximum scale – minSize * minScale < size < maxSize * maxScale. 0.85 if excluded or None	NA	Float	float
SIFTsteps	SIFT steps per scale octave. 3 if excluded or None	NA	Integer	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
pairJson	JSON file where tile pairs are stored (.json, .gz, .zip)	(REQUIRED)	InputFile	str
pbs_template	pbs template to wrap spark job	(REQUIRED)	InputFile	str
no_nodes	Number of nodes to run the pbs job	30	Integer	int
ppn	Number of processors per node (default = 30)	30	Integer	int
queue_name	Name of the queue to submit the job	connectome	String	str
logdir	location to set logging for qsub command	(REQUIRED)	OutputDir	str

opts = <marshmallow.schema.SchemaOpts object>

```
class asap.pointmatch.schemas.PointMatchClientParametersSpark(extra=None, only=None,
                                                                exclude=(), prefix="", strict=None,
                                                                many=False, context=None,
                                                                load_only=(), dump_only=(),
                                                                partial=False)
```

Bases: [SparkParameters](#), [SIFTPointMatchParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 63: PointMatchClientParametersSpark

key	description	default	field_type	json_type
baseDataUrl	api endpoint url e.g. <a href="http://<host>[:port]/render-ws/v1">http://<host>[:port]/render-ws/v1	NA	String	str
owner	owner of match collection	NA	String	str
collection	match collection name	NA	String	str
render	no description	NA	RenderClientParameters	
matchRod	Ratio of first to second nearest neighbors used as a cutoff in matching features. 0.92 if excluded or None	NA	Float	float
matchModel-Type	Model to match for RANSAC filtering. 'AFFINE' if excluded or None	NA	String	str
matchIterations	RANSAC filter iterations. 1000 if excluded or None	NA	Integer	int
matchMax-Epsilon	no description	NA	Float	float
matchMinInlierRatio	Minimal ratio of inliers to candidates for successful RANSAC filtering. 0.0 if excluded or None	NA	Float	float
matchMinNumInliers	Minimum absolute number of inliers for successful RANSAC filtering. 4 if excluded or None	NA	Integer	int
matchMaxNumInliers	Maximum absolute number of inliers allowed after RANSAC filtering. unlimited if excluded or None	NA	Integer	int
matchMax-Trust	Maximum trust for filtering such that candidates with cost larger than matchMax-Trust * median cost are rejected. 3.0 if excluded or None	NA	Float	float
matchFilter	whether to match one set of matches, or multiple sets. And, whether to keep them separate, or aggregate them. SINGLE_SET if excluded or None.	NA	String	str
rootFeatureDirectory	Root directory for saved feature lists. Features extracted from dynamically rendered canvases if excluded or None.	NA	String	str
require-StoredFeatures	Whether to throw an exception in case features stored in rootFeatureDirectory cannot be found. Missing features are extracted from dynamically rendered canvases if excluded or None	NA	Boolean	bool
maxFeature-CacheGb	Maximum size of feature cache, in GB. 2GB if excluded or None	NA	Integer	int

continues on next page

Table 63 – continued from previous page

key	description	default	field_type	json_type
clipWidth	Full scale pixels to include in clipped rendering of LEFT/RIGHT oriented tile pairs. Will not LEFT/RIGHT clip if excluded or None.	NA	Integer	int
clipHeight	Full scale pixels to include in clipped rendering of TOP/BOTTOM oriented tile pairs. Will not TOP/BOTTOM clip if excluded or None.	NA	Integer	int
renderScale	Scale at which image tiles will be rendered. 1.0 (full scale) if excluded or None	NA	Float	float
renderWith-Filter	Render tiles using default filtering (0 and 255 pixel values replaced with integer in U(64, 191), followed by default NormalizeLocalContrast). True if excluded or None	NA	Boolean	bool
renderWith-outMask	Render tiles without mipMapLevel masks. True if excluded or None	NA	Boolean	bool
render-FullScaleWidth	Full scale width for all rendered tiles	NA	Integer	int
render-FullScale-Height	Full scale height for all rendered tiles	NA	Integer	int
fillWithNoise	Fill each canvas image with noise prior to rendering. True if excluded or None	NA	Boolean	bool
renderFilterListName	Apply specified filter list to all renderings	NA	String	str
SIFTfdSize	SIFT feature descriptor size – samples per row and column. 8 if excluded or None	NA	Integer	int
SIFTmin-Scale	SIFT minimum scale – minSize * minScale < size < maxSize * maxScale. 0.5 if excluded or None	NA	Float	float
SIFTmaxScale	SIFT maximum scale – minSize * minScale < size < maxSize * maxScale. 0.85 if excluded or None	NA	Float	float
SIFTsteps	SIFT steps per scale octave. 3 if excluded or None	NA	Integer	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
pairJson	JSON file where tile pairs are stored (.json, .gz, .zip)	(REQUIRED)	InputFile	str
jarfile	spark jar to call java spark command	(REQUIRED)	String	str
className	spark class to call	(REQUIRED)	String	str
driverMemory	spark driver memory (important for local spark)	6g	String	str
memory	Memory required for spark job	NA	String	str
sparkhome	Spark home directory containing bin/spark_submit	(REQUIRED)	InputDir	str

continues on next page

Table 63 – continued from previous page

key	description	default	field_type	json_type
spark_files	list of spark files to add to the spark submit command	NA	List	str
spark_conf	dictionary of key value pairs to add to spark_submit as --conf key=value	NA	Dict	?
masterUrl	spark master url. For local execution local[num_procs,num_retries]	(REQUIRED)	String	str

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.pointmatch.schemas.PointMatchOpenCVParameters(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: [RenderParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 64: PointMatchOpenCVParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(REQUIRED)	RenderClientParameters	Dict
ndiv	one tile per tile pair subdivided into ndiv x ndiv for easier homography finding	8	Integer	int
matchMax	per tile pair limit, randomly chosen after SIFT and RANSAC	1000	Integer	int
downsample_scale	passed to cv2.resize(fx=, fy=)	0.3	Float	float
SIFT_nfeature	passed to cv2.xfeatures2d.SIFT_create(nfeatures=)	20000	Integer	int
SIFT_noctave	passed to cv2.xfeatures2d.SIFT_create(nOctaveLayers=)	3	Integer	int
SIFT_sigma	passed to cv2.xfeatures2d.SIFT_create(sigma=)	1.5	Float	float
RANSAC_outlier	passed to cv2.findHomography(src, dst, cv2.RANSAC, outlier)	5.0	Float	float
FLANN_ntree	passed to cv2.FlannBasedMatcher()	5	Integer	int
FLANN_ncheck	passed to cv2.FlannBasedMatcher()	50	Integer	int
ratio_of_dist	ratio in Lowe's ratio test	0.7	Float	float
CLAHE_grid	tileGridSize for cv2 CLAHE	None	Integer	int
CLAHE_clip	clipLimit for cv2 CLAHE	None	Float	float
pairJson	full path of tilepair json	NA	String	str
input_stack	Name of raw input lens data stack	NA	String	str
match_collection	name of point match collection	NA	String	str
ncpus	number of CPUs to use	-1	Integer	int

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.pointmatch.schemas.SIFTPointMatchParameters(extra=None, only=None, exclude=(),
                                                         prefix="", strict=None, many=False,
                                                         context=None, load_only=(), dump_only=(),
                                                         partial=False)
```

Bases: [ArgSchema](#), [FeatureExtractionParameters](#), [FeatureRenderParameters](#),
[FeatureRenderClipParameters](#), [FeatureStorageParameters](#), [MatchDerivationParameters](#),
[RenderParametersMatchWebServiceParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 65: SIFTPointMatchParameters

key	description	default	field_type	json_type
baseDataUrl	api endpoint url e.g. http://<host>[:port]/render-ws/v1	NA	String	str
owner	owner of match collection	NA	String	str
collection	match collection name	NA	String	str
render	no description	NA	RenderClientParameters	
matchRod	Ratio of first to second nearest neighbors used as a cutoff in matching features. 0.92 if excluded or None	NA	Float	float
matchModel-Type	Model to match for RANSAC filtering. 'AFFINE' if excluded or None	NA	String	str
matchIterations	RANSAC filter iterations. 1000 if excluded or None	NA	Integer	int
matchMax-Epsilon	no description	NA	Float	float
matchMinInlierRatio	Minimal ratio of inliers to candidates for successful RANSAC filtering. 0.0 if excluded or None	NA	Float	float
matchMinNumInliers	Minimum absolute number of inliers for successful RANSAC filtering. 4 if excluded or None	NA	Integer	int
matchMaxNumInliers	Maximum absolute number of inliers allowed after RANSAC filtering. unlimited if excluded or None	NA	Integer	int
matchMax-Trust	Maximum trust for filtering such that candidates with cost larger than matchMax-Trust * median cost are rejected. 3.0 if excluded or None	NA	Float	float
matchFilter	whether to match one set of matches, or multiple sets. And, whether to keep them separate, or aggregate them. SINGLE_SET if excluded or None.	NA	String	str
rootFeatureDirectory	Root directory for saved feature lists. Features extracted from dynamically rendered canvases if excluded or None.	NA	String	str
require-StoredFeatures	Whether to throw an exception in case features stored in rootFeatureDirectory cannot be found. Missing features are extracted from dynamically rendered canvases if excluded or None	NA	Boolean	bool

continues on next page

Table 65 – continued from previous page

key	description	default	field_type	json_type
maxFeature-CacheGb	Maximum size of feature cache, in GB. 2GB if excluded or None	NA	Integer	int
clipWidth	Full scale pixels to include in clipped rendering of LEFT/RIGHT oriented tile pairs. Will not LEFT/RIGHT clip if excluded or None.	NA	Integer	int
clipHeight	Full scale pixels to include in clipped rendering of TOP/BOTTOM oriented tile pairs. Will not TOP/BOTTOM clip if excluded or None.	NA	Integer	int
renderScale	Scale at which image tiles will be rendered. 1.0 (full scale) if excluded or None	NA	Float	float
renderWith-Filter	Render tiles using default filtering (0 and 255 pixel values replaced with integer in U(64, 191), followed by default NormalizeLocalContrast). True if excluded or None	NA	Boolean	bool
renderWith-outMask	Render tiles without mipMapLevel masks. True if excluded or None	NA	Boolean	bool
render-FullScaleWidth	Full scale width for all rendered tiles	NA	Integer	int
render-FullScale-Height	Full scale height for all rendered tiles	NA	Integer	int
fillWithNoise	Fill each canvas image with noise prior to rendering. True if excluded or None	NA	Boolean	bool
renderFilterListName	Apply specified filter list to all renderings	NA	String	str
SIFTfdSize	SIFT feature descriptor size – samples per row and column. 8 if excluded or None	NA	Integer	int
SIFTmin-Scale	SIFT minimum scale – minSize * minScale < size < maxSize * maxScale. 0.5 if excluded or None	NA	Float	float
SIFTmaxScale	SIFT maximum scale – minSize * minScale < size < maxSize * maxScale. 0.85 if excluded or None	NA	Float	float
SIFTsteps	SIFT steps per scale octave. 3 if excluded or None	NA	Integer	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
pairJson	JSON file where tile pairs are stored (.json, .gz, .zip)	(REQUIRED)	InputFile	str

opts = <marshmallow.schema.SchemaOpts object>

```
class asap.pointmatch.schemas.SwapPointMatches(extra=None, only=None, exclude=(), prefix="",
strict=None, many=False, context=None,
load_only=(), dump_only=(), partial=False)
```

Bases: [RenderParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 66: SwapPointMatches

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
match_owner	Match collection owner name	(RE-REQUIRED)	String	str
source_collection	Source point match collection	(RE-REQUIRED)	String	str
target_collection	Target point match collection	(RE-REQUIRED)	String	str
zValues	List of integer group ids	(RE-REQUIRED)	List	int
pool_size	Pool size	5	Integer	int

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.pointmatch.schemas.SwapPointMatchesOutput(extra=None, only=None, exclude=(), prefix="",
strict=None, many=False, context=None,
load_only=(), dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 67: SwapPointMatchesOutput

key	description	default	field_type	json_type
source_collection	Source point match collection	(RE-REQUIRED)	String	str
target_collection	Target point match collection	(RE-REQUIRED)	String	str
swapped_zs	List of group ids that got swapped	(RE-REQUIRED)	List	int
non-swapped_zs	List of group ids that did not get swapped	(RE-REQUIRED)	List	int

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.pointmatch.schemas.TilePairClientOutputParameters(extra=None, only=None, exclude=(),
prefix="", strict=None, many=False,
context=None, load_only=(),
dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 68: TilePairClientOutputParameters

key	description	default	field_type	json_type
tile_pair_file	location of json file with tile pair inputs	(RE-REQUIRED)	InputFile	str

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.pointmatch.schemas.TilePairClientParameters(extra=None, only=None, exclude=(),
                                                         prefix="", strict=None, many=False,
                                                         context=None, load_only=(), dump_only=(),
                                                         partial=False)
```

Bases: [RenderParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 69: TilePairClientParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
stack	input stack to which tilepairs need to be generated	(RE-REQUIRED)	String	str
baseStack	Base stack	None	String	str
minZ	z min for generating tilepairs	None	Integer	int
maxZ	z max for generating tilepairs	None	Integer	int
xyNeighbor-Factor	Multiply this by max(width, height) of each tile to determine radius for locating neighbor tiles	0.9	Float	float
zNeighbor-Distance	Look for neighbor tiles with z values less than or equal to this distance from the current tile's z value	2	Integer	int
excludeCornerNeighbors	Exclude neighbor tiles whose center x and y is outside the source tile's x and y range respectively	True	Boolean	bool
exclude-SameLayerNeighbors	Exclude neighbor tiles in the same layer (z) as the source tile	False	Boolean	bool
excludeCompletelyObscuredTiles	Exclude tiles that are completely obscured by reacquired tiles	True	Boolean	bool
output_dir	Output directory path to save the tilepair json file	(RE-REQUIRED)	OutputDir	str
memGB	Memory for the java client to run	6G	String	str

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
validate_data(data)
```

Module contents

asap.residuals package

Submodules

asap.residuals.compute_residuals module

asap.residuals.compute_residuals.**compute_mean_tile_residuals**(*residuals*)

asap.residuals.compute_residuals.**compute_residuals_within_group**(*render*, *stack*,
matchCollectionOwner,
matchCollection, *z*,
min_points=1, *tilspecs*=None)

asap.residuals.compute_residuals.**get_tile_centers**(*tilspecs*)

Module contents

asap.rough_align package

Submodules

asap.rough_align.apply_rough_alignment_to_montages module

exception asap.rough_align.apply_rough_alignment_to_montages.**ApplyRoughAlignmentException**
Bases: [*RenderModuleException*](#)

Something is wrong in ApplyRough....

class asap.rough_align.apply_rough_alignment_to_montages.**ApplyRoughAlignmentTransform**(*schema_type*=None,
**args*,
***kwargs*)

Bases: [*RenderModule*](#)

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type [*ApplyRoughAlignmentTransformParameters*](#)

default_output_schema

alias of [*ApplyRoughAlignmentOutputParameters*](#)

default_schema

alias of [*ApplyRoughAlignmentTransformParameters*](#)

run()

asap.rough_align.apply_rough_alignment_to_montages.**add_masks_to_lowres**(*render*, *stack*, *z*,
mask_map)

```
asap.rough_align.apply_rough_alignment_to_montages.apply_rough_alignment(render, input_stack,
                                                                    prealigned_stack,
                                                                    lowres_stack,
                                                                    output_stack,
                                                                    output_dir, scale,
                                                                    mask_input_dir, up-
                                                                    date_lowres_with_masks,
                                                                    read_masks_from_lowres_stack,
                                                                    fil-
                                                                    ter_montage_output_with_masks,
                                                                    mask_exts, Z,
                                                                    apply_scale=False,
                                                                    consolidateTrans-
                                                                    forms=True,
                                                                    remap_section_ids=False)

asap.rough_align.apply_rough_alignment_to_montages.filter_highres_with_masks(resolved_highres,
                                                                    tspec_lowres,
                                                                    mask_map)
```

function to return a filtered list of tilespecs from a

ResolvedTiles object, based on a lowres mask

Parameters

- **resolved_highres** (*renderapi.resolvedtiles.ResolvedTiles* object) – tile-specs and transforms from a single section
- **tspec_lowres** (*renderapi.tilespec.TileSpec* object) – tilespec from a downsampled stack
- **mask_map** (*dict*) – keys should match lowres tileids, values are mask file URI

Returns

new_highres – which highres specs are fully contained within the boundary of mask=255

Return type

List of *renderapi.tilespec.TileSpec* objects

```
asap.rough_align.apply_rough_alignment_to_montages.get_mask_paths(mask_input_dir, tilespecs,
                                                                    read_masks_from_lowres_stack,
                                                                    exts=['png', 'tif'])
```

asap.rough_align.do_rough_alignment module

asap.rough_align.schemas module

```
class asap.rough_align.schemas.ApplyRoughAlignmentOutputParameters(extra=None, only=None,
                                                                    exclude=(), prefix="",
                                                                    strict=None, many=False,
                                                                    context=None, load_only=(),
                                                                    dump_only=(),
                                                                    partial=False)
```

Bases: [DefaultSchema](#)

Table 70: ApplyRoughAlignmentOutputParameters

key	description	default	field_type	json_type
zs	list of z values that were applied to	NA	NumpyArray	?
output_stack	stack where applied transforms were set	NA	String	str

opts = `<marshmallow.schema.SchemaOpts object>`

class `asap.rough_align.schemas.ApplyRoughAlignmentTransformParameters` (*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: [RenderParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 71: ApplyRoughAlignmentTransformParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
montage_stack	stack to make a downsample version of	(RE-REQUIRED)	String	str
pre-aligned_stack	stack with dropped tiles corrected for stitching errors	None	String	str
lowres_stack	montage scape stack with rough aligned transform	(RE-REQUIRED)	String	str
output_stack	output high resolution rough aligned stack	(RE-REQUIRED)	String	str
tile-spec_directory	path to save section images	(RE-REQUIRED)	String	str
map_z	map the montage Z indices to the rough alignment indices (default - False)	False	Boolean	bool
remap_section_ids	map section ids as well with the new z mapping. Default = False	False	Boolean	bool
consolidate_transforms	should the transforms be consolidated? (default - True)	True	Boolean	bool
scale	scale of montage scapes	(RE-REQUIRED)	Float	float
apply_scale	do you want to apply scale	False	Boolean	bool
pool_size	pool size for parallel processing	10	Integer	int
new_z	List of new z values to be mapped to	None	List	int
old_z	List of z values to apply rough alignment to	(RE-REQUIRED)	List	int
mask_input_dir	directory containing mask files. base-names of masks that match tileIds in the rough stack will be handled.	None	InputDir	str
read_masks_from_lowres_stack	whether masks will be taken from lowres tilespecs. any mask_input_dir ignored.	False	Boolean	bool
update_lowres_with_masks	should the masks be added to the rough stack	False	Boolean	bool
filter_montage_output_with_masks	should the tiles written be filtered by the masks	False	Boolean	bool
mask_exts	what kind of mask files to recognize	['png', 'tif']	List	str
close_stack	whether to set output stack to COMPLETE	True	Boolean	bool

`opts = <marshmallow.schema.SchemaOpts object>`

`validate_data(data)`

```
class asap.rough_align.schemas.DownsamplingMaskHandlerSchema(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: [RenderParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 72: DownsampleMaskHandlerSchema

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
stack	stack that is read and modified with masks	(RE-REQUIRED)	String	str
close_stack	set COMPLETE or not	True	Boolean	bool
mask_dir	directory containing masks, named <z>_*.png where<z> is a z value in input_stack and may be specified in z_apply	None	InputDir	str
collection	name of collection to be filtered by mask, or reset can be None for no operation	None	String	str
zMask	z values for which the masks will be set	None	List	int
zReset	z values for which the masks will be reset	None	List	int
mask_exts	what kind of mask files to recognize	['png', 'tif']	List	str

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.rough_align.schemas.LowresStackParameters(extra=None, only=None, exclude=(), prefix="",
strict=None, many=False, context=None,
load_only=(), dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 73: LowresStackParameters

key	description	default	field_type	json_type
stack	Input downsample images section stack	(RE-REQUIRED)	String	str
owner	Owner of the input lowres stack	None	String	str
project	Project of the input lowres stack	None	String	str
service_host	Service host for the input stack Render service	None	String	str
baseURL	Base URL of the Render service for the source stack	None	String	str
renderbin-Path	Client scripts location	None	String	str
verbose	Want the output to be verbose?	0	Integer	int

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.rough_align.schemas.MakeAnchorStackSchema(extra=None, only=None, exclude=(), prefix="",
strict=None, many=False, context=None,
load_only=(), dump_only=(), partial=False)
```

Bases: [StackTransitionParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 74: MakeAnchorStackSchema

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	not used in this module, keeps parents happy	[1000]	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
input_stack	no description	(RE-REQUIRED)	String	str
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
over-write_zlayer	no description	False	Boolean	bool
output_stackVersion	no description	NA	RenderStackValidation	
transform_xml	xml transforms from trakemimages from which these are made are assumed to be named <z>_*.png	NA	InputFile	str
transform_json	Human generated list of transforms.or, json scraped from xmlKeys are of form <z>_*.png where z matches a tilespec in input_stack and values are AffineModel transform jsonswill override xml input.	None	InputFile	str

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.rough_align.schemas.OutputLowresStackParameters(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 75: OutputLowresStackParameters

key	description	default	field_type	json_type
stack	Input downsample images section stack	(RE-REQUIRED)	String	str
owner	Owner of the input lowres stack	None	String	str
project	Project of the input lowres stack	None	String	str
service_host	Service host for the input stack Render service	None	String	str
baseURL	Base URL of the Render service for the source stack	None	String	str
renderbin-Path	Client scripts location	None	String	str
verbose	Want the output to be verbose?	0	Integer	int

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.rough_align.schemas.PairwiseRigidOutputSchema(extra=None, only=None, exclude=(),
    prefix="", strict=None, many=False,
    context=None, load_only=(),
    dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 76: PairwiseRigidOutputSchema

key	description	default	field_type	json_type
minZ	minimum z value in output stack	(RE-REQUIRED)	Integer	int
maxZ	maximum z value in output stack	(RE-REQUIRED)	Integer	int
output_stack	name of output stack	(RE-REQUIRED)	String	str
missing	list of z values missing in z range of output stack	(RE-REQUIRED)	List	int
masked	list of z values masked in z range of output stack	(RE-REQUIRED)	List	int
residuals	pairwise residuals in output stack	(RE-REQUIRED)	List	?

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.rough_align.schemas.PairwiseRigidSchema(extra=None, only=None, exclude=(), prefix="",
    strict=None, many=False, context=None,
    load_only=(), dump_only=(), partial=False)
```

Bases: [StackTransitionParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 77: PairwiseRigidSchema

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
input_stack	no description	(RE-REQUIRED)	String	str
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
over-write_zlayer	no description	False	Boolean	bool
output_stackVersion	no description	NA	RenderStackValidation	
match_collection	Point match collection name	(RE-REQUIRED)	String	str
gap_file	json file {k: v} where int(k) is a z value to skip entries in here that are not already missing will be omitted from the output stack i.e. this is a place one can skip sections	None	InputFile	str
translate_to_positive	translate output stack to positive space	True	Boolean	bool
translation_buffer	minimum (x, y) of output stack if translate_to_positive=True	[0, 0]	List	float
anchor_stack	fix transforms using tiles in this stack	None	String	str

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.rough_align.schemas.PointMatchCollectionParameters(extra=None, only=None,
                                                                exclude=(), prefix="", strict=None,
                                                                many=False, context=None,
                                                                load_only=(), dump_only=(),
                                                                partial=False)
```

Bases: [DefaultSchema](#)

Table 78: PointMatchCollectionParameters

key	description	default	field_type	json_type
owner	Point match collection owner (defaults to render owner)	None	String	str
match_collection	Point match collection name	(RE-REQUIRED)	String	str
server	baseURL of the Render service holding the point match collection	None	String	str
verbose	Verbose output flag	0	Integer	int

```
opts = <marshmallow.schema.SchemaOpts object>
```

Module contents

asap.stack package

Submodules

asap.stack consolidate_transforms module

```
class asap.stack.consolidate_transforms.ConsolidateTransforms(schema_type=None, *args,
                                                             **kwargs)
```

Bases: *RenderModule*

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type *ConsolidateTransformsParameters*

default_output_schema

alias of *ConsolidateTransformsOutputParameters*

default_schema

alias of *ConsolidateTransformsParameters*

run()

```
asap.stack.consolidate_transforms.consolidate_transforms(tforms, ref_tforms=[],
                                                         logger=<RootLogger root (WARNING)>,
                                                         makePolyDegree=0,
                                                         keep_ref_tforms=False)
```

```
asap.stack.consolidate_transforms.dereference_tforms(tforms, ref_tforms)
```

```
asap.stack.consolidate_transforms.flatten_and_dereference_tforms(tforms, ref_tforms)
```

```
asap.stack.consolidate_transforms.flatten_tforms(tforms)
```

```
asap.stack.consolidate_transforms.process_z(render, stack, outstack, transform_slice, z)
```

asap.stack.redirect_mipmaps module

change storage directory of imageUrl in a given mipMapLevel

```
class asap.stack.redirect_mipmaps.RedirectMipMapsModule(schema_type=None, *args, **kwargs)
```

Bases: *StackTransitionModule*

Note: This class takes a ArgSchema as an input to parse inputs , with a default schema of type *RedirectMipMapsParameters*

default_output_schema

alias of *RedirectMipMapsOutput*

default_schema

alias of *RedirectMipMapsParameters*

static `get_replacement_ImagePyramid(ip, mml_d_map)`

run()

asap.stack.schemas module

class `asap.stack.schemas.ConsolidateTransformsOutputParameters`(*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: *DefaultSchema*

Table 79: ConsolidateTransformsOutputParameters

key	description	default	field_type	json_type
output_stack	name of output stack	(RE-REQUIRED)	<i>String</i>	str
numZ	Number of z values processed	(RE-REQUIRED)	<i>Integer</i>	int

opts = `<marshmallow.schema.SchemaOpts object>`

class `asap.stack.schemas.ConsolidateTransformsParameters`(*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: *RenderParameters*

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 80: ConsolidateTransformsParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
stack	stack to consolidate	(RE-REQUIRED)	String	str
postfix	postfix to add to stack name on saving if no output defined (default _CONS)	_CONS	String	str
trans-forms_slice	a string representing a slice describing the set of transforms to be consolidated (i.e. 1:)	slice(None, None, None)	Slice	str
output_stack	name of output stack (default to adding postfix to input)	NA	String	str
pool_size	name of output stack (default to adding postfix to input)	10	Integer	int
minZ	minimum z to consolidate in read in from stack and write to output_stack. Default to minimum z in stack	NA	Float	float
maxZ	maximum z to consolidate in read in from stack and write to output_stack. Default to maximum z in stack	NA	Float	float
over-write_zlayer	whether to remove the existing layer from the target stack before uploading.	False	Boolean	bool
close_stack	no description	False	Boolean	bool

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.stack.schemas.MipMapDirectories(extra=None, only=None, exclude=(), prefix="", strict=None,
many=False, context=None, load_only=(), dump_only=(),
partial=False)
```

Bases: [DefaultSchema](#)

Table 81: MipMapDirectories

key	description	default	field_type	json_type
level	mipMapLevel for which parent directory will be changed	(RE-REQUIRED)	Integer	int
directory	directory where relocated mipmaps are found.	(RE-REQUIRED)	InputDir	str

opts = `<marshmallow.schema.SchemaOpts object>`

```
class asap.stack.schemas.RedirectMipMapsOutput(extra=None, only=None, exclude=(), prefix="",
strict=None, many=False, context=None,
load_only=(), dump_only=(), partial=False)
```

Bases: [DefaultSchema](#)

Table 82: RedirectMipMapsOutput

key	description	default	field_type	json_type
zValues	no description	(RE-REQUIRED)	List	int
output_stack	no description	(RE-REQUIRED)	String	str

opts = <marshmallow.schema.SchemaOpts object>

class asap.stack.schemas.RedirectMipMapsParameters(*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: [StackTransitionParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 83: RedirectMipMapsParameters

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
input_stack	no description	(RE-REQUIRED)	String	str
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
over-write_zlayer	no description	False	Boolean	bool
output_stackVersion	no description	NA	RenderStackVersion	
new_mipmap_directories	no description	(RE-REQUIRED)	MipMapDirectories	

opts = <marshmallow.schema.SchemaOpts object>

class asap.stack.schemas.RemapZsOutput(*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: [DefaultSchema](#)

Table 84: RemapZsOutput

key	description	default	field_type	json_type
zValues	no description	(RE-REQUIRED)	List	int
output_stack	no description	(RE-REQUIRED)	String	str

opts = <marshmallow.schema.SchemaOpts object>

class asap.stack.schemas.RemapZsParameters(*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: [StackTransitionParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 85: RemapZsParameters

key	description	default	field_type	json_type
pool_size	no description	1	Integer	int
minZ	no description	NA	Integer	int
maxZ	no description	NA	Integer	int
z	no description	NA	Integer	int
zValues	no description	NA	List	int
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(RE-REQUIRED)	RenderClientParameters	
input_stack	no description	(RE-REQUIRED)	String	str
output_stack	no description	(RE-REQUIRED)	String	str
close_stack	no description	False	Boolean	bool
over-write_zlayer	no description	False	Boolean	bool
output_stackVersion	no description	NA	RenderStackVersion	
remap_sectionId	no description	NA	Boolean	bool
new_zValues	no description	(RE-REQUIRED)	List	int

opts = <marshmallow.schema.SchemaOpts object>

class asap.stack.schemas.SwapZsOutput(*extra=None, only=None, exclude=(), prefix="", strict=None, many=False, context=None, load_only=(), dump_only=(), partial=False*)

Bases: [DefaultSchema](#)

Table 86: SwapZsOutput

key	description	default	field_type	json_type
source_stacks	List of source stacks that have been successfully swapped	(REQUIRED)	List	str
target_stacks	List of target stacks that have been successfully swapped	(REQUIRED)	List	str
swapped_zvalues	no description	NA	List	?

`opts = <marshmallow.schema.SchemaOpts object>`

```
class asap.stack.schemas.SwapZsParameters(extra=None, only=None, exclude=(), prefix="", strict=None,
many=False, context=None, load_only=(), dump_only=(),
partial=False)
```

Bases: [RenderParameters](#)

This schema is designed to be a schema_type for an ArgSchemaParser object

Table 87: SwapZsParameters

key	description	default	field_type	json_type
input_json	file path of input json file	NA	InputFile	str
output_json	file path to output json file	NA	OutputFile	str
log_level	set the logging level of the module	ERROR	LogLevel	str
render	parameters to connect to render server	(REQUIRED)	RenderClientParameters	
source_stack	List of source stacks	(REQUIRED)	List	str
target_stack	List of target stacks	(REQUIRED)	List	str
complete_source_stack	set source stack state to complete after copying Default=False	False	Boolean	bool
complete_target_stack	set target stack state to complete after copying Default=False	False	Boolean	bool
zValues	no description	NA	List	?
delete_source_stack	Do you want to delete source stack after copying its contents?. Default=False	False	Boolean	bool
pool_size	Pool size	5	Integer	int

`opts = <marshmallow.schema.SchemaOpts object>`

Module contents

7.1.2 Module contents

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

a

asap, 88
asap.dataimport, 23
asap.dataimport.apply_mipmaps_to_render, 13
asap.dataimport.create_mipmaps, 14
asap.dataimport.generate_EM_tilespecs_from_metadata, 15
asap.dataimport.generate_mipmaps, 15
asap.dataimport.make_montage_scapes_stack, 16
asap.dataimport.schemas, 16
asap.em_montage_qc, 28
asap.em_montage_qc.detect_montage_defects, 23
asap.em_montage_qc.plots, 24
asap.em_montage_qc.schemas, 24
asap.intensity_correction, 32
asap.intensity_correction.apply_multiplicative_correction, 28
asap.intensity_correction.calculate_multiplicative_correction, 30
asap.intensity_correction.schemas, 30
asap.lens_correction, 35
asap.lens_correction.apply_lens_correction, 32
asap.lens_correction.schemas, 33
asap.materialize, 41
asap.materialize.materialize_sections, 35
asap.materialize.render_downsample_sections, 36
asap.materialize.schemas, 36
asap.module, 58
asap.module.render_module, 56
asap.module.schemas, 56
asap.module.schemas.renderclient_schemas, 41
asap.module.schemas.schemas, 49
asap.module.schemas.spark_schemas, 50
asap.module.schemas.stack_schemas, 51
asap.module.template_module, 57
asap.point_match_optimization, 63
asap.point_match_optimization.schemas, 58
asap.pointmatch, 75
asap.pointmatch.create_tilepairs, 63
asap.pointmatch.generate_point_matches_qsub, 63
asap.pointmatch.generate_point_matches_spark, 64
asap.pointmatch.schemas, 65
asap.residuals, 75
asap.residuals.compute_residuals, 75
asap.rough_align, 83
asap.rough_align.apply_rough_alignment_to_montages, 75
asap.rough_align.schemas, 76
asap.stack, 88
asap.stack consolidate_transforms, 83
asap.stack.redirect_mipmaps, 83
asap.stack.schemas, 84

INDEX

A

add_arg() (in module `asap.dataimport.apply_mipmaps_to_render` module, 13
asap.pointmatch.generate_point_matches_spark), 64
add_masks_to_lowres() (in module `asap.dataimport.create_mipmaps` module, 14
asap.rough_align.apply_rough_alignment_to_montages), 75
add_match_collection_owner() (in module `asap.dataimport.generate_EM_tilespecs_from_metafile` module, 15
asap.rough_align.apply_rough_alignment_to_montages), 75
add_mipmaps_to_render() (in module `asap.dataimport.generate_mipmaps` module, 15
asap.dataimport.apply_mipmaps_to_render), 13
AddMipMapsToStack (class in module `asap.dataimport.make_montage_scapes_stack` module, 16
asap.dataimport.apply_mipmaps_to_render), 13
AddMipMapsToStackOutput (class in module `asap.dataimport.schemas` module, 16
asap.dataimport.schemas), 16
AddMipMapsToStackParameters (class in module `asap.em_montage_qc` module, 28
asap.dataimport.schemas), 17
apply_rough_alignment() (in module `asap.em_montage_qc.detect_montage_defects` module, 23
asap.rough_align.apply_rough_alignment_to_montages), 75
ApplyLensCorrection (class in module `asap.em_montage_qc.plots` module, 24
asap.lens_correction.apply_lens_correction), 32
ApplyLensCorrectionOutput (class in module `asap.em_montage_qc.schemas` module, 24
asap.lens_correction.schemas), 33
ApplyLensCorrectionParameters (class in module `asap.intensity_correction` module, 32
asap.lens_correction.schemas), 33
ApplyRoughAlignmentException, 75
ApplyRoughAlignmentOutputParameters (class in module `asap.intensity_correction.apply_multiplicative_correction` module, 28
asap.rough_align.schemas), 76
ApplyRoughAlignmentTransform (class in module `asap.intensity_correction.calculate_multiplicative_correction` module, 30
asap.rough_align.apply_rough_alignment_to_montages), 75
ApplyRoughAlignmentTransformParameters (class in module `asap.intensity_correction.schemas` module, 30
asap.rough_align.schemas), 77
asap module, 88
asap.dataimport module, 23
asap.dataimport.apply_mipmaps_to_render module, 13
asap.dataimport.create_mipmaps module, 14
asap.dataimport.generate_EM_tilespecs_from_metafile module, 15
asap.dataimport.generate_mipmaps module, 15
asap.dataimport.make_montage_scapes_stack module, 16
asap.dataimport.schemas module, 16
asap.em_montage_qc module, 28
asap.em_montage_qc.detect_montage_defects module, 23
asap.em_montage_qc.plots module, 24
asap.em_montage_qc.schemas module, 24
asap.intensity_correction module, 32
asap.intensity_correction.apply_multiplicative_correction module, 28
asap.intensity_correction.calculate_multiplicative_correction module, 30
asap.intensity_correction.schemas module, 30
asap.lens_correction module, 35
asap.lens_correction.apply_lens_correction module, 32
asap.lens_correction.schemas module, 33
asap.materialize module, 41
asap.materialize.materialize_sections module, 35
asap.materialize.render_downsample_sections module, 36
asap.materialize.schemas module, 36

asap.module
 module, 58
 asap.module.render_module
 module, 56
 asap.module.schemas
 module, 56
 asap.module.schemas.renderclient_schemas
 module, 41
 asap.module.schemas.schemas
 module, 49
 asap.module.schemas.spark_schemas
 module, 50
 asap.module.schemas.stack_schemas
 module, 51
 asap.module.template_module
 module, 57
 asap.point_match_optimization
 module, 63
 asap.point_match_optimization.schemas
 module, 58
 asap.pointmatch
 module, 75
 asap.pointmatch.create_tilepairs
 module, 63
 asap.pointmatch.generate_point_matches_qsub
 module, 63
 asap.pointmatch.generate_point_matches_spark
 module, 64
 asap.pointmatch.schemas
 module, 65
 asap.residuals
 module, 75
 asap.residuals.compute_residuals
 module, 75
 asap.rough_align
 module, 83
 asap.rough_align.apply_rough_alignment_to_montages
 module, 75
 asap.rough_align.schemas
 module, 76
 asap.stack
 module, 88
 asap.stack consolidate_transforms
 module, 83
 asap.stack.redirect_mipmaps
 module, 83
 asap.stack.schemas
 module, 84

B

Bounds (*class in asap.materialize.schemas*), 36

C

check_stack_for_mipmaps() (in module

asap.materialize.render_downsample_sections), 36

check_status_of_stack() (in module *asap.em_montage_qc.detect_montage_defects*), 23

client_class(*asap.pointmatch.create_tilepairs.TilePairClientModule* attribute), 63

client_script_name(*asap.pointmatch.create_tilepairs.TilePairClientModule* attribute), 63

CollectionId (*class in asap.pointmatch.schemas*), 65

compute_mean_tile_residuals() (in module *asap.residuals.compute_residuals*), 75

compute_residuals_within_group() (in module *asap.residuals.compute_residuals*), 75

consolidate_transforms() (in module *asap.stack.consolidate_transforms*), 83

ConsolidateTransforms (*class in asap.stack.consolidate_transforms*), 83

ConsolidateTransformsOutputParameters (*class in asap.stack.schemas*), 84

ConsolidateTransformsParameters (*class in asap.stack.schemas*), 84

create_mipmap_from_tuple() (in module *asap.dataimport.generate_mipmaps*), 15

create_mipmap_from_tuple_uri() (in module *asap.dataimport.generate_mipmaps*), 15

create_mipmaps() (in module *asap.dataimport.create_mipmaps*), 14

create_mipmaps_legacy() (in module *asap.dataimport.create_mipmaps*), 14

create_mipmaps_uri() (in module *asap.dataimport.create_mipmaps*), 14

create_montage_scape_tile_specs() (in module *asap.dataimport.make_montage_scapes_stack*), 16

create_tilespecs_without_mipmaps() (in module *asap.materialize.render_downsample_sections*), 36

CreateMipMapException, 14

D

default_output_schema (*asap.dataimport.apply_mipmaps_to_render.AddMipMapsToStack* attribute), 13

default_output_schema (*asap.dataimport.generate_EM_tilespecs_from_metafile.GenerateEMTilespecs* attribute), 15

default_output_schema (*asap.dataimport.generate_mipmaps.GenerateMipMaps* attribute), 15

default_output_schema (*asap.dataimport.make_montage_scapes_stack.MakeMontageScapesStack* attribute), 16

default_output_schema

[illegible]

class method), 36
DownsampleMaskHandlerSchema (class in
asap.rough_align.schemas), 78

F

FeatureExtractionParameters (class in
asap.module.schemas.renderclient_schemas),
41
FeatureRenderClipParameters (class in
asap.module.schemas.renderclient_schemas),
42
FeatureRenderParameters (class in
asap.module.schemas.renderclient_schemas),
42
FeatureStorageParameters (class in
asap.module.schemas.renderclient_schemas),
43
filter_highres_with_masks() (in module
asap.rough_align.apply_rough_alignment_to_montages),
76
fix_badkey() (*asap.module.schemas.stack_schemas.OverridableParameterSchema*
static method), 53
flatten_and_dereference_tforms() (in module
asap.stack consolidate_transforms), 83
flatten_tforms() (in module
asap.stack consolidate_transforms), 83
form_sift_params_list() (in module
asap.pointmatch.generate_point_matches_spark),
64

G

generate_zValues() (*asap.module.schemas.stack_schemas.ZValueParameterSchema*
method), 55
GenerateEMTileSpecsModule (class in
asap.dataimport.generate_EM_tilespecs_from_metafile),
15
GenerateEMTileSpecsOutput (class in
asap.dataimport.schemas), 17
GenerateEMTileSpecsParameters (class in
asap.dataimport.schemas), 18
GenerateMipMaps (class in
asap.dataimport.generate_mipmaps), 15
GenerateMipMapsOutput (class in
asap.dataimport.schemas), 19
GenerateMipMapsParameters (class in
asap.dataimport.schemas), 20
get_args() (*asap.materialize.materialize_sections.MaterializeSectionsModule*
class method), 35
get_args() (*asap.module.render_module.SparkModule*
class method), 56
get_args() (*asap.pointmatch.generate_point_matches_spark.PointMatchClientModuleSpark*
class method), 64
get_cmd_opt() (*asap.module.render_module.SparkModule*
static method), 56

get_filepath_from_tilespec() (in module
asap.dataimport.generate_mipmaps), 16
get_flag_cmd() (*asap.module.render_module.SparkModule*
static method), 56
get_host_port_dict_from_url() (in module
asap.pointmatch.generate_point_matches_spark),
64
get_inputstack_zs() (in module
asap.module.render_module.StackInputModule
method), 57
get_mask_paths() (in module
asap.rough_align.apply_rough_alignment_to_montages),
76
get_materialize_options() (in module
asap.materialize.materialize_sections.MaterializeSectionsModule
class method), 35
get_overlapping_inputstack_zvalues() (in module
asap.module.render_module.StackInputModule
method), 57
get_pointmatch_args() (in module
asap.pointmatch.generate_point_matches_spark.PointMatchClientModuleSpark
class method), 64
get_pre_post_tspecs() (in module
asap.em_montage_qc.detect_montage_defects),
24
get_replacement_ImagePyramid() (in module
asap.stack.redirect_mipmaps.RedirectMipMapsModule
static method), 84
get_spark_call() (*asap.module.render_module.SparkModule*
class method), 56
get_spark_command() (in module
asap.module.render_module.SparkModule
class method), 56
get_tile_centers() (in module
asap.residuals.compute_residuals), 75
getImage() (in module
asap.intensity_correction.apply_multiplicative_correction),
28
getImageFromTilespecs() (in module
asap.intensity_correction.calculate_multiplicative_correction),
30
|
image_coords_from_stage() (in module
asap.dataimport.generate_EM_tilespecs_from_metafile.GenerateEMTileSpecsModule
static method), 15
image_directory_to_prefix() (in module
asap.dataimport.schemas.GenerateEMTileSpecsParameters
method), 19
input_stacks (*asap.em_montage_qc.schemas.DetectDistortionParameters*
attribute), 25
InputStackParameters (class in
asap.module.schemas.stack_schemas), 51
intensity_corr() (in module

- asap.intensity_correction.apply_multiplicative_correction* (in module *asap.intensity_correction*), 29
- mipmap_block_reduce()* (in module *asap.dataimport.create_mipmaps*), 14
- mipmap_directory_to_prefix()* (*asap.dataimport.schemas.AddMipMapsToStackParameters* method), 17
- mipmap_directories* (class in *asap.stack.schemas*), 85
- mipmap_directories* module
- LowresStackParameters* (class in *asap*, 88
- asap.rough_align.schemas*), 79
- asap.dataimport*, 23
- asap.dataimport.apply_mipmaps_to_render*, 13
- asap.dataimport.create_mipmaps*, 14
- asap.dataimport.generate_EM_tilespecs_from_metafile*, 15
- asap.dataimport.generate_mipmaps*, 15
- asap.dataimport.make_montage_scapes_stack*, 16
- asap.dataimport.schemas*, 16
- asap.em_montage_qc*, 28
- asap.em_montage_qc.detect_montage_defects*, 23
- asap.em_montage_qc.plots*, 24
- asap.em_montage_qc.schemas*, 24
- asap.intensity_correction*, 32
- asap.intensity_correction.apply_multiplicative_correction*, 28
- asap.intensity_correction.calculate_multiplicative_correction*, 30
- asap.intensity_correction.schemas*, 30
- asap.lens_correction*, 35
- asap.lens_correction.apply_lens_correction*, 32
- asap.lens_correction.schemas*, 33
- asap.materialize*, 41
- asap.materialize.materialize_sections*, 35
- asap.materialize.render_downsample_sections*, 36
- asap.materialize.schemas*, 36
- asap.module*, 58
- asap.module.render_module*, 56
- asap.module.schemas*, 56
- asap.module.schemas.renderclient_schemas*, 41
- asap.module.schemas.schemas*, 49
- asap.module.schemas.spark_schemas*, 50
- asap.module.schemas.stack_schemas*, 51
- asap.module.template_module*, 57
- asap.point_match_optimization*, 63
- asap.point_match_optimization.schemas*, 58
- asap.pointmatch*, 75
- asap.pointmatch.create_tilepairs*, 63
- asap.pointmatch.generate_point_matches_qsub*, 63
- asap.pointmatch.generate_point_matches_spark*, 64
- asap.pointmatch.schemas*, 65
- asap.residuals*, 75
- L**
- M**
- make_median_image()* (in module *asap.intensity_correction.calculate_multiplicative_correction*), 30
- make_tilespecs_and_cmds()* (in module *asap.dataimport.generate_mipmaps*), 16
- MakeAnchorStackSchema* (class in *asap.rough_align.schemas*), 79
- MakeMedian* (class in *asap.intensity_correction.calculate_multiplicative_correction*), 30
- MakeMedianParams* (class in *asap.intensity_correction.schemas*), 30
- MakeMontageScapeSectionStack* (class in *asap.dataimport.make_montage_scapes_stack*), 16
- MakeMontageScapeSectionStackOutput* (class in *asap.dataimport.schemas*), 21
- MakeMontageScapeSectionStackParameters* (class in *asap.dataimport.schemas*), 21
- maskUrl_to_uri()* (*asap.dataimport.schemas.GenerateEMTileSpecsParameters* method), 19
- maskUrl_to_uri()* (*asap.lens_correction.schemas.ApplyLensCorrectionParameters* method), 34
- MatchDerivationParameters* (class in *asap.module.schemas.renderclient_schemas*), 44
- MatchWebServiceParameters* (class in *asap.module.schemas.renderclient_schemas*), 45
- MaterializedBoxParameters* (class in *asap.module.schemas.renderclient_schemas*), 45
- MaterializeSectionsError*, 35
- MaterializeSectionsModule* (class in *asap.materialize.materialize_sections*), 35
- MaterializeSectionsOutput* (class in *asap.materialize.schemas*), 37
- MaterializeSectionsParameters* (class in *asap.materialize.schemas*), 37
- metafile_to_uri()* (*asap.dataimport.schemas.GenerateEMTileSpecsParameters* method), 19
- mipmap_block_reduce()* (in module *asap.dataimport.create_mipmaps*), 14
- mipmap_directory_to_prefix()* (*asap.dataimport.schemas.AddMipMapsToStackParameters* method), 17

[asap.residuals.compute_residuals](#), 75
[asap.rough_align](#), 83
[asap.rough_align.apply_rough_alignment_to_montages](#), 75
[asap.rough_align.schemas](#), 76
[asap.stack](#), 88
[asap.stack consolidate_transforms](#), 83
[asap.stack.redirect_mipmaps](#), 83
[asap.stack.schemas](#), 84
[MultIntensityCorr](#) (class in [asap.intensity_correction.apply_multiplicative_correction](#)), 28
[MultIntensityCorrParams](#) (class in [asap.intensity_correction.schemas](#)), 31
O
[opts \(asap.dataimport.schemas.AddMipMapsToStackOutput attribute\)](#), 17
[opts \(asap.dataimport.schemas.AddMipMapsToStackParameters attribute\)](#), 17
[opts \(asap.dataimport.schemas.GenerateEMTileSpecsOutput attribute\)](#), 18
[opts \(asap.dataimport.schemas.GenerateEMTileSpecsParameters attribute\)](#), 19
[opts \(asap.dataimport.schemas.GenerateMipMapsOutput attribute\)](#), 20
[opts \(asap.dataimport.schemas.GenerateMipMapsParameters attribute\)](#), 20
[opts \(asap.dataimport.schemas.MakeMontageScapeSectionStackOutput attribute\)](#), 21
[opts \(asap.dataimport.schemas.MakeMontageScapeSectionStackParameters attribute\)](#), 22
[opts \(asap.em_montage_qc.schemas.DetectDistortionParameters attribute\)](#), 25
[opts \(asap.em_montage_qc.schemas.DetectDistortionParametersOutput attribute\)](#), 25
[opts \(asap.em_montage_qc.schemas.DetectMontageDefectsParameters attribute\)](#), 26
[opts \(asap.em_montage_qc.schemas.DetectMontageDefectsParametersOutput attribute\)](#), 27
[opts \(asap.em_montage_qc.schemas.RoughQCOutputSchema attribute\)](#), 27
[opts \(asap.em_montage_qc.schemas.RoughQCSchema attribute\)](#), 28
[opts \(asap.intensity_correction.schemas.MakeMedianParams attribute\)](#), 31
[opts \(asap.intensity_correction.schemas.MultIntensityCorrParams attribute\)](#), 32
[opts \(asap.lens_correction.schemas.ApplyLensCorrectionOutput attribute\)](#), 33
[opts \(asap.lens_correction.schemas.ApplyLensCorrectionParameters attribute\)](#), 34
[opts \(asap.lens_correction.schemas.TransformParameters attribute\)](#), 34
[opts \(asap.materialize.schemas.Bounds attribute\)](#), 36
[opts \(asap.materialize.schemas.DeleteMaterializedSectionsOutput attribute\)](#), 37
[opts \(asap.materialize.schemas.DeleteMaterializedSectionsParameters attribute\)](#), 37
[opts \(asap.materialize.schemas.MaterializeSectionsOutput attribute\)](#), 37
[opts \(asap.materialize.schemas.MaterializeSectionsParameters attribute\)](#), 39
[opts \(asap.materialize.schemas.RenderSectionAtScaleOutput attribute\)](#), 39
[opts \(asap.materialize.schemas.RenderSectionAtScaleParameters attribute\)](#), 40
[opts \(asap.materialize.schemas.ValidateMaterializationOutput attribute\)](#), 40
[opts \(asap.materialize.schemas.ValidateMaterializationParameters attribute\)](#), 41
[opts \(asap.module.schemas.renderclient_schemas.FeatureExtractionParameters attribute\)](#), 42
[opts \(asap.module.schemas.renderclient_schemas.FeatureRenderClipParameters attribute\)](#), 42
[opts \(asap.module.schemas.renderclient_schemas.FeatureRenderParameters attribute\)](#), 43
[opts \(asap.module.schemas.renderclient_schemas.FeatureStorageParameters attribute\)](#), 44
[opts \(asap.module.schemas.renderclient_schemas.MatchDerivationParameters attribute\)](#), 45
[opts \(asap.module.schemas.renderclient_schemas.MatchWebServiceParameters attribute\)](#), 45
[opts \(asap.module.schemas.renderclient_schemas.MaterializedBoxParameters attribute\)](#), 46
[opts \(asap.module.schemas.renderclient_schemas.RenderParametersMatch attribute\)](#), 47
[opts \(asap.module.schemas.renderclient_schemas.RenderParametersRender attribute\)](#), 47
[opts \(asap.module.schemas.renderclient_schemas.RenderWebServiceParameters attribute\)](#), 48
[opts \(asap.module.schemas.renderclient_schemas.WebServiceParameters attribute\)](#), 48
[opts \(asap.module.schemas.renderclient_schemas.ZRangeParameters attribute\)](#), 48
[opts \(asap.module.schemas.schemas.RenderClientParameters attribute\)](#), 49
[opts \(asap.module.schemas.schemas.RenderParameters attribute\)](#), 49
[opts \(asap.module.schemas.schemas.TemplateOutputParameters attribute\)](#), 50
[opts \(asap.module.schemas.schemas.TemplateParameters attribute\)](#), 50
[opts \(asap.module.schemas.spark_schemas.SparkOptions attribute\)](#), 50
[opts \(asap.module.schemas.spark_schemas.SparkParameters attribute\)](#), 51
[opts \(asap.module.schemas.stack_schemas.InputStackParameters attribute\)](#), 51

[attribute](#)), 52
[opts \(asap.module.schemas.stack_schemas.OutputStackParameters attribute\)](#), 52
[opts \(asap.module.schemas.stack_schemas.OverridableParameterSchema attribute\)](#), 53
[opts \(asap.module.schemas.stack_schemas.ProcessPoolParameters attribute\)](#), 53
[opts \(asap.module.schemas.stack_schemas.RenderCycle attribute\)](#), 53
[opts \(asap.module.schemas.stack_schemas.RenderMipMapParameters attribute\)](#), 53
[opts \(asap.module.schemas.stack_schemas.RenderStackVersion attribute\)](#), 54
[opts \(asap.module.schemas.stack_schemas.StackTransitionParameters attribute\)](#), 55
[opts \(asap.module.schemas.stack_schemas.ZValueParameters attribute\)](#), 55
[opts \(asap.point_match_optimization.schemas.PointMatchOptions attribute\)](#), 59
[opts \(asap.point_match_optimization.schemas.PointMatchOptions attribute\)](#), 59
[opts \(asap.point_match_optimization.schemas.PtMatchOptions attribute\)](#), 60
[opts \(asap.point_match_optimization.schemas.PtMatchOptions attribute\)](#), 61
[opts \(asap.point_match_optimization.schemas.SIFT_options attribute\)](#), 62
[opts \(asap.point_match_optimization.schemas.url_options attribute\)](#), 63
[opts \(asap.pointmatch.schemas.CollectionId attribute\)](#), 65
[opts \(asap.pointmatch.schemas.PointMatchClientOutputSchema attribute\)](#), 65
[opts \(asap.pointmatch.schemas.PointMatchClientParametersQsub attribute\)](#), 67
[opts \(asap.pointmatch.schemas.PointMatchClientParametersSpark attribute\)](#), 70
[opts \(asap.pointmatch.schemas.PointMatchOpenCVParameters attribute\)](#), 70
[opts \(asap.pointmatch.schemas.SIFTPointMatchParameters attribute\)](#), 72
[opts \(asap.pointmatch.schemas.SwapPointMatches attribute\)](#), 73
[opts \(asap.pointmatch.schemas.SwapPointMatchesOutput attribute\)](#), 73
[opts \(asap.pointmatch.schemas.TilePairClientOutputParameters attribute\)](#), 73
[opts \(asap.pointmatch.schemas.TilePairClientParameters attribute\)](#), 74
[opts \(asap.rough_align.schemas.ApplyRoughAlignmentOutputParameters attribute\)](#), 77
[opts \(asap.rough_align.schemas.ApplyRoughAlignmentTransformParameters attribute\)](#), 78
[opts \(asap.rough_align.schemas.DownsampleMaskHandlerSchema attribute\)](#), 79
[opts \(asap.rough_align.schemas.LowresStackParameters attribute\)](#), 79
[opts \(asap.rough_align.schemas.MakeAnchorStackSchema attribute\)](#), 80
[opts \(asap.rough_align.schemas.OutputLowresStackParameters attribute\)](#), 81
[opts \(asap.rough_align.schemas.PairwiseRigidOutputSchema attribute\)](#), 81
[opts \(asap.rough_align.schemas.PairwiseRigidSchema attribute\)](#), 82
[opts \(asap.rough_align.schemas.PointMatchCollectionParameters attribute\)](#), 83
[opts \(asap.stack.schemas.ConsolidateTransformsOutputParameters attribute\)](#), 84
[opts \(asap.stack.schemas.ConsolidateTransformsParameters attribute\)](#), 85
[opts \(asap.stack.schemas.MipMapDirectories attribute\)](#), 85
[opts \(asap.stack.schemas.RedirectMipMapsOutput attribute\)](#), 86
[opts \(asap.stack.schemas.RedirectMipMapsParameters attribute\)](#), 86
[opts \(asap.stack.schemas.RemapZsOutput attribute\)](#), 87
[opts \(asap.stack.schemas.RemapZsParameters attribute\)](#), 87
[opts \(asap.stack.schemas.SwapZsOutput attribute\)](#), 88
[opts \(asap.stack.schemas.SwapZsParameters attribute\)](#), 88
[output_tilespecs_to_stack\(\) \(asap.module.render_module.StackOutputModule method\)](#), 57
[OutputLowresStackParameters \(class in asap.rough_align.schemas\)](#), 80
[OutputStackParameters \(class in asap.module.schemas.stack_schemas\)](#), 52
[OverridableParameterSchema \(class in asap.module.schemas.stack_schemas\)](#), 52
[override_input\(\) \(asap.module.schemas.stack_schemas.OverridableParameterSchema method\)](#), 53

P

[PairwiseRigidOutputSchema \(class in asap.rough_align.schemas\)](#), 81
[PairwiseRigidSchema \(class in asap.rough_align.schemas\)](#), 81
[plot_defects\(\) \(in asap.em_montage_qc.plots\)](#), 24
[plot_residual\(\) \(in asap.em_montage_qc.plots\)](#), 24
[plot_section_maps\(\) \(in asap.em_montage_qc.plots\)](#), 24
[point_match_plot\(\) \(in asap.em_montage_qc.plots\)](#), 24

[PointMatchClientModuleQsub](#) (class in [asap.pointmatch.generate_point_matches_qsub](#)), 63
[PointMatchClientModuleSpark](#) (class in [asap.pointmatch.generate_point_matches_spark](#)), 64
[PointMatchClientOutputSchema](#) (class in [asap.pointmatch.schemas](#)), 65
[PointMatchClientParametersQsub](#) (class in [asap.pointmatch.schemas](#)), 65
[PointMatchClientParametersSpark](#) (class in [asap.pointmatch.schemas](#)), 67
[PointMatchCollectionParameters](#) (class in [asap.rough_align.schemas](#)), 82
[PointMatchOpenCVParameters](#) (class in [asap.pointmatch.schemas](#)), 70
[PointMatchOptimizationParameters](#) (class in [asap.point_match_optimization.schemas](#)), 58
[PointMatchOptimizationParametersOutput](#) (class in [asap.point_match_optimization.schemas](#)), 59
[process_tile\(\)](#) (in module [asap.intensity_correction.apply_multiplicative_correction](#)), 29
[process_z\(\)](#) (in module [asap.stack consolidate_transforms](#)), 83
[ProcessPoolParameters](#) (class in [asap.module.schemas.stack_schemas](#)), 53
[PtMatchOptimizationParameters](#) (class in [asap.point_match_optimization.schemas](#)), 59
[PtMatchOptimizationParametersOutput](#) (class in [asap.point_match_optimization.schemas](#)), 60

R

[randomly_subsample_tilespecs\(\)](#) (in module [asap.intensity_correction.calculate_multiplicative_correction](#)), 30
[RedirectMipMapsModule](#) (class in [asap.stack.redirect_mipmaps](#)), 83
[RedirectMipMapsOutput](#) (class in [asap.stack.schemas](#)), 85
[RedirectMipMapsParameters](#) (class in [asap.stack.schemas](#)), 86
[RemapZsOutput](#) (class in [asap.stack.schemas](#)), 86
[RemapZsParameters](#) (class in [asap.stack.schemas](#)), 87
[RenderClientParameters](#) (class in [asap.module.schemas.schemas](#)), 49
[RenderCycle](#) (class in [asap.module.schemas.stack_schemas](#)), 53
[RenderMipMapPathBuilder](#) (class in [asap.module.schemas.stack_schemas](#)), 53
[RenderModule](#) (class in [asap.module.render_module](#)), 56
[RenderModuleException](#), 56
[RenderParameters](#) (class in [asap.module.schemas.schemas](#)), 49
[RenderParametersMatchWebServiceParameters](#) (class in [asap.module.schemas.renderclient_schemas](#)), 46
[RenderParametersRenderWebServiceParameters](#) (class in [asap.module.schemas.renderclient_schemas](#)), 47
[RenderSectionAtScale](#) (class in [asap.materialize.render_downsample_sections](#)), 36
[RenderSectionAtScaleOutput](#) (class in [asap.materialize.schemas](#)), 39
[RenderSectionAtScaleParameters](#) (class in [asap.materialize.schemas](#)), 39
[RenderStackVersion](#) (class in [asap.module.schemas.stack_schemas](#)), 54
[RenderWebServiceParameters](#) (class in [asap.module.schemas.renderclient_schemas](#)), 48
[RoughQCOutputSchema](#) (class in [asap.em_montage_qc.schemas](#)), 27
[RoughQCSchema](#) (class in [asap.em_montage_qc.schemas](#)), 27
[run\(\)](#) ([asap.dataimport.apply_mipmaps_to_render.AddMipMapsToStack](#) method), 13
[run\(\)](#) ([asap.dataimport.generate_EM_tilespecs_from_metafile.GenerateEM](#) method), 15
[run\(\)](#) ([asap.dataimport.generate_mipmaps.GenerateMipMaps](#) method), 15
[run\(\)](#) ([asap.dataimport.make_montage_scapes_stack.MakeMontageScapes](#) method), 16
[run\(\)](#) ([asap.em_montage_qc.detect_montage_defects.DetectMontageDefects](#) method), 23
[run\(\)](#) ([asap.intensity_correction.apply_multiplicative_correction.MultiInter](#) method), 28
[run\(\)](#) ([asap.intensity_correction.calculate_multiplicative_correction.Make](#) method), 30
[run\(\)](#) ([asap.lens_correction.apply_lens_correction.ApplyLensCorrection](#) method), 33
[run\(\)](#) ([asap.materialize.materialize_sections.MaterializeSectionsModule](#) method), 35
[run\(\)](#) ([asap.materialize.render_downsample_sections.RenderSectionAtScale](#) method), 36
[run\(\)](#) ([asap.module.template_module.TemplateModule](#) method), 57
[run\(\)](#) ([asap.pointmatch.create_tilepairs.TilePairClientModule](#) method), 63
[run\(\)](#) ([asap.pointmatch.generate_point_matches_qsub.PointMatchClientM](#) method), 63
[run\(\)](#) ([asap.pointmatch.generate_point_matches_spark.PointMatchClientM](#) method), 64
[run\(\)](#) ([asap.rough_align.apply_rough_alignment_to_montages.ApplyRough](#) method), 75

run() (*asap.stack consolidate_transforms.ConsolidateTransformPairClientModule* (class in *asap.pointmatch.create_tilepairs*), 63
 method), 83
 run() (*asap.stack.redirect_mipmaps.RedirectMipMapsModule* (class in *asap.pointmatch.schemas*), 73
 method), 84
 run_analysis() (in module *TilePairClientParameters* (class in *asap.pointmatch.schemas*), 73
asap.em_montage_qc.detect_montage_defects), 24
 TransformParameters (class in *asap.lens_correction.schemas*), 34
 run_spark_command() (*asap.module.render_module.SparkModule* (class in *ts_from_imgdata*) (*asap.dataimport.generate_EM_tilespecs_from_metafile*) (class in *asap.dataimport.generate_EM_tilespecs_from_metafile*) (method), 15
 method), 56

S

sanitize_cmd() (*asap.module.render_module.SparkModule* (class in *asap.point_match_optimization.schemas*),
 static method), 56
 sectionId_from_z() (*asap.dataimport.generate_EM_tilespecs_from_metafile.GenerateEMTileSpecsModule* (class in
 static method), 15

SIFT_options (class in *asap.point_match_optimization.schemas*), 61

SIFTPointMatchParameters (class in *asap.pointmatch.schemas*), 70

SparkModule (class in *asap.module.render_module*), 56

SparkModuleError, 56

SparkOptions (class in *asap.module.schemas.spark_schemas*), 50

SparkParameters (class in *asap.module.schemas.spark_schemas*), 51

StackInputModule (class in *asap.module.render_module*), 56

StackOutputModule (class in *asap.module.render_module*), 57

StackTransitionModule (class in *asap.module.render_module*), 57

StackTransitionParameters (class in *asap.module.schemas.stack_schemas*), 54

SwapPointMatches (class in *asap.pointmatch.schemas*), 72

SwapPointMatchesOutput (class in *asap.pointmatch.schemas*), 73

SwapZsOutput (class in *asap.stack.schemas*), 87

SwapZsParameters (class in *asap.stack.schemas*), 88

T

TemplateModule (class in *asap.module.template_module*), 57

TemplateOutputParameters (class in *asap.module.schemas.schemas*), 49

TemplateParameters (class in *asap.module.schemas.schemas*), 50

threshold_cutoff (*asap.em_montage_qc.schemas.DetectDirectionalEdges* (class in *asap.em_montage_qc.schemas*),
 attribute), 25

tileId_from_basename() (*asap.dataimport.generate_EM_tilespecs_from_metafile.GenerateEMTileSpecsModule* (class in *asap.dataimport.generate_EM_tilespecs_from_metafile*) (method), 15
 module method), 15

U

url_options (class in *asap.point_match_optimization.schemas*), 62

V

validate_data() (*asap.dataimport.schemas.MakeMontageScapeSectionS* (class in *asap.dataimport.schemas*) (method), 22

validate_data() (*asap.materialize.schemas.RenderSectionAtScaleParameters* (class in *asap.materialize.schemas*) (method), 40

validate_data() (*asap.point_match_optimization.schemas.PtMatchOptimizationParameters* (class in *asap.pointmatch_optimization.schemas*) (method), 60

validate_data() (*asap.pointmatch.schemas.TilePairClientParameters* (class in *asap.pointmatch.schemas*) (method), 74

validate_data() (*asap.rough_align.schemas.ApplyRoughAlignmentTransformParameters* (class in *asap.rough_align.schemas*) (method), 78

validate_options() (*asap.module.schemas.renderclient_schemas.RenderClientOptions* (class in *asap.module.schemas.renderclient_schemas*) (method), 47

validate_options() (*asap.module.schemas.renderclient_schemas.RenderClientOptions* (class in *asap.module.schemas.renderclient_schemas*) (method), 48

validate_tilespecs() (*asap.module.render_module.StackOutputModule* (class in *asap.module.render_module*) (method), 57

ValidateMaterializationOutput (class in *asap.materialize.schemas*), 40

ValidateMaterializationParameters (class in *asap.materialize.schemas*), 40

validationOptions() (*asap.dataimport.schemas.GenerateMipMapsParameters* (class in *asap.dataimport.schemas*) (method), 21

W

WebServiceParameters (class in *asap.module.schemas.renderclient_schemas*), 48

WithThreadPool (class in *asap.materialize.render_downsample_sections*), 36

write_image() (*asap.dataimport.generate_EM_tilespecs_from_metafile.GenerateEMTileSpecsModule* (class in *asap.dataimport.generate_EM_tilespecs_from_metafile*) (method), 15
 module method), 15

Z

ZRangeParameters (class in
asap.module.schemas.renderclient_schemas),
[48](#)

ZValueParameters (class in
asap.module.schemas.stack_schemas), [55](#)